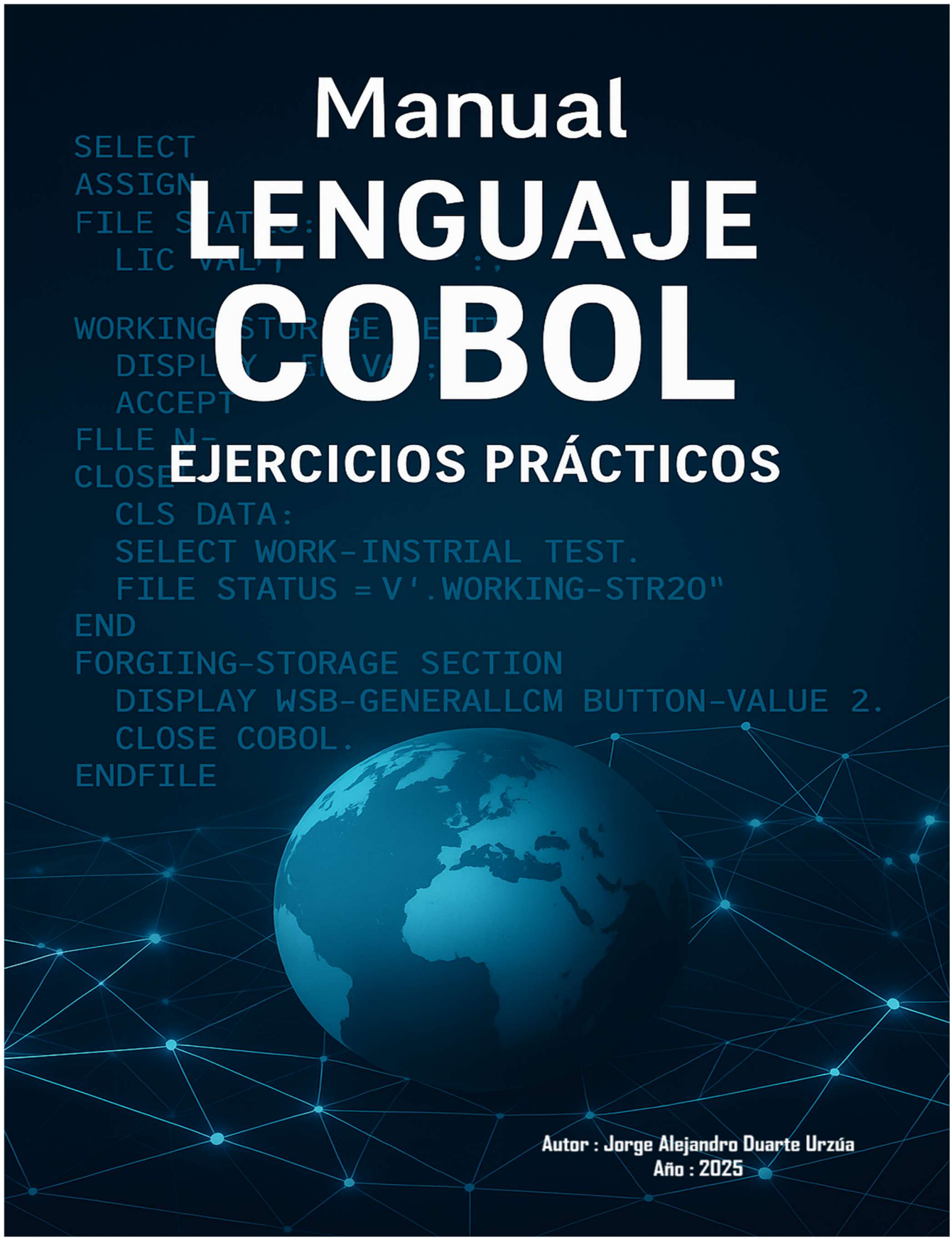
Manual

LENGUAJE COBOL

EJERCICIOS PRÁCTICOS



Autor : Jorge Alejandro Duarte Urzúa

Año : 2025

Manual Lenguaje COBOL Ejercicios Prácticos

NO COPIAR

Autor : Jorge Alejandro Duarte Urzúa

Año : 2025

Contacto : Jorge.duarte,urzua@gmail.com

linkedin.com/in/jorge-duarte-urzúa-75532043

Contenido

Sobre el Autor 6
Introducción 7
Historia Lenguaje Cobol 8
El origen del lenguaje de programación COBOL 8
COBOL ha sido todo un éxito sin precedentes desde su creación 9
¿Por qué COBOL sigue siendo relevante tras más de 60 años? 10
Conociendo el Lenguaje Cobol 12
1. IDENTIFICATION DIVISION 12
2. ENVIRONMENT DIVISION 12
3. DATA DIVISION 12
4. PROCEDURE DIVISION 12
5. Hojas de Codificación 13
6. Palabras Reservadas 14
7. File Status 17
Fuentes del Manual 19
Instalación OpenCobolIde 19
Primer Programa Cobol (HolaMundo) 23
Programas Útiles 25
Menú Básico (MENU.cbl) 25
Cálcula Digito Verificador Rut (CalculaDVRut.CBL) 27
Valida Rut (ValidaRut.CBL) 29
Probar ValidaRut y CalulaDVRut (PruebaValidaRut.CBL) 31
Calcular Largo String Máx 500 Caracteres (LargoString.CBL) 33
Convertir String en Mayúsculas (ConvertirMayusculas.CBL) 34
Convertir String en Minúsculas (ConvertirMinusculas.CBL) 35

NO COPIAR

Prueba Convertir en Mayúsculas/Minúsculas (PruebaMayusculasMinusculas.CBL) 36
Elimina los Blancos Inciales y Finales (Trim.CBL) 38
Convierte String en SnakeCase (SnakeCase.CBL) 39
Probar Trim y SnakeCase (ProbarTrimySnake.CBL) 40
Validar una Fecha (ValidarFecha.CBL) 42
Sumar días a una Fecha (SumarDiasFecha.CBL) 44
Probar Sumar días a una Fecha (ProbarSumarDiasFecha.CBL) 47
Diferencia Dias entre 2 Fechas (DiffDiasFecha.CBL) 49
Probar Diferencia Dias entre 2 Fechas (ProbarDiffFechas.CBL) 51
Calcular Largo String Máx 10000 Caracteres (LargoString10000.CBL) 53
Validar Tarjeta de Crédito (ValidaTarjetaCredito.CBL) 54
Calcular Edad (CalcularEdad.CBL) 56
Buscar Posición de un Carácter (PosCarStr.CBL) 58
Centrar Texto (CentrarTexto.CBL) 59
Validar Email (ValidarEmail.CBL) 60
Probar Validar Email (ProbarEmail.CBL) 62
Leer un Archivo (LeerArchivo.CBL) 64
Mensaje (VentanaMsj.CBL) 66
Grabar en un Archivo (GrabarCliente.CBL) 68
Borrar un Registro de un Archivo (BorrarCliente.CBL) 71
Consultar un Registro de un Archivo (ConsultaCliente.CBL) 73
Mantenedor de un Cliente (Mantenedor.CBL) 76
Listado de Registros 1 (ListadoClientes.CBL) 84
Listado de Registros 2 (ListadoClientes2.CBL) 93
Generar Archivo para Calendario (GenCalendario.CBL) 97
Mostrar Fecha Calendario (FechasCalendario.CBL) 99
Mostrar Calendario (MostrarCalendario.CBL) 103
Bonus Track AS400 108
Creación Archivo de Clientes (CLIENTES-PF) 109

NO COPIAR

Creación Pantalla Consulta de Clientes (CONS01) 110
Creación Pantalla Listado de Clientes (LIST01) 112
Creación Pantalla Mantenedor de Clientes (MANT01) 113
Creación Pantalla Menú (MENU01) 114
Creación Pantalla Visualizar Mensajes (VENMSG) 115
Creación Visualizar Mensajes (VERMSJ.CBL) 117
Largo String (LARGOSTR.CBL) 119
Validar Rut (VALRUT.CBL) 120
Validar Fecha (VALFEC.CBL) 122
Grabar Cliente (GRABARCLI.CBL) 124
Borrar Cliente (BORRARCLI.CBL) 125
Mantenedor de Cliente (MANCLICBL.CBL) 127
Listado de Cliente (LISCLI.CBL) 133
Programa Menu (MENUCLP) 136
Ejecución Programas AS400 137
Referencias 144

NO COPIAR

Sobre el Autor

Jorge Alejandro Duarte Urzúa, nació en Rancagua el 11 de Enero de 1972, en CHILE, desarrollo su formación escolar en la escuela F227 de Olivar Alto, posteriormente por el trabajo de su Padre, se mudaron a Rancagua donde desarrollo la mayor parte de su vida escolar.

Una vez terminada su educación secundaria, entra a estudiar al Instituto de Capacitación Profesional (INACAP), donde cursa la carrera de Análisis de Sistemas, que a los 2 Años permitía obtener el Título de Programador de Aplicaciones Computacionales.

En 2014, continúa sus estudios de Ingeniería de Ejecución de Informática, obteniendo además la Licenciatura en Ciencias de la Informática, ambas en la Universidad de Las Américas.

En el ámbito profesional se ha desarrollado en diversas empresas y rubros, trabajando para empresas de Seguros, Banca y Retail, durante los más de 30 años de experiencia a desarrollado una capacidad de abstracción y análisis que permiten trabajar no solo en diversos temas y lenguajes.

Como se comprenderá la mayor parte laboral ha sido desarrollando en Mainframe AS400, donde desarrolla con COBOL y RPG, en diversas versiones, además de trabajar con Colas MQ (WebSphere MQ Series) y con el Toolkit Websphere Message Broker, hoy ACE12 con el cual se pueden realizar Integraciones SOA.

NO COPIAR

Introducción

El presente Manual se desarrolla para que personas que NO tienen conocimiento del lenguaje, así como los que ya poseen, puedan experimentar y practicar lo básico del Lenguaje.

Se presentará a grandes rasgos lo esencial del Lenguaje COBOL, su origen, historia, esto para entender su origen y porque hasta el día de hoy sigue siendo importante tanto a nivel Gubernamental, como en las grandes corporaciones.

Pero no sólo hablaremos de Historia, **el principal motivo de desarrollar este Manual es tener Código Funcional de Código y Rutinas de uso común o que puedan llegar a ser necesarias en algún momento**, y que puede ser utilizado en cualquier ambiente, el código que se presentará está desarrollado con **OpenCobolIde**, existirán programas básicos, con manejo de Archivo, Rutinas, Consultas.

Además se incluirá un Capitulo relacionado al Manejo de Cobol en ambiente ISeries ó cómo muchos lo conocen AS400, existirá pantallas, Programas CLP, Archivos y Programas Cobol y manejo de Cobol con SQL.

NO COPIAR

Historia Lenguaje Cobol

COBOL es el acrónimo de COmmon Business-Oriented Language, traducido como "Lenguaje Común Orientado a Negocios". Se trata de un lenguaje de programación creado en 1959. Hace más de 60 años. Y aunque debería haber caído en el olvido o ser parte de la historia de la informática, empresas y programadores siguen usándolo. Hasta el punto de situarse entre los 20 lenguajes de programación más demandados a día de hoy.

¿Qué hace que un lenguaje tan longevo siga vivo después de tanto tiempo? Una simple búsqueda en Internet nos devuelve multitud de artículos en los que se destaca que COBOL no está muerto. Y que está vivo. Pero los motivos no están tan claros. ¿Es por su simplicidad? ¿Por su versatilidad? ¿Su longevidad es lo que lo ha convertido en un lenguaje de programación a prueba de todo?

Una de las claves podemos encontrarlas en una encuesta publicada el año pasado por el vendedor de software Micro Focus. Especializado en aplicaciones y herramientas para empresas, en su encuesta, COBOL salía muy bien parado. Incluso más de lo esperado. El 92% de los encuestados consideraban este lenguaje de programación como "estratégico". Y prueba de ello son las más de 800.000 líneas de código escritas con COBOL que siguen usándose a diario en empresas de todo el mundo.

El origen del lenguaje de programación COBOL

Antes de repasar los motivos por los que este lenguaje veterano sigue siendo clave en computadoras y empresas, hagamos un breve repaso a su nacimiento. COBOL nace gracias a un ambicioso proyecto en el que participaron fabricantes de computadoras, empresas y organismos usuarios de las mismas y el Departamento de Defensa de Estados Unidos. En mayo de 1959 crearon la comisión CODASYL, acrónimo de Conference on Data Systems Languages. Traducido, Conferencia sobre Lenguajes de Sistemas de Datos. En ella estaban representados nombres pesados como los fabricantes IBM y RCA o la Fuerza Aérea. Así como otros fabricantes de computadoras ya caídos en el olvido y otras agencias estadounidenses que, en la práctica, eran los principales clientes y usuarios de esos ordenadores.

¿Por qué esa unión tan prometedora? En la década de 1950, las computadoras eran gigantes metálicos que ocupan mesas, armarios y, en algunos casos, gran parte de las salas o habitaciones de oficinas, empresas y organismos públicos. Máquinas de grandes proporciones, muy caras y difíciles de gestionar. Limitadas a las empresas, las universidades, los centros de investigación y organismos públicos con grandes presupuestos.

NO COPIAR

Uno de los motivos de la dificultad de manejar estas computadoras eran sus lenguajes de programación. La interacción entre máquina y humano se hacía con códigos de programación. Y en la década de 1950, surgen varios lenguajes de programación como FLOW-MATIC, FORTRAN, AIMACO, ALGOL o COMTRAN. Incompatibles entre sí. Adaptar programas y sus códigos a cada lenguaje tenía un coste económico. Y de ahí el proyecto que daría pie a COBOL: crear un lenguaje común para todas las computadoras de la época.

COBOL ha sido todo un éxito sin precedentes desde su creación

Aunque lo más fácil hubiera sido adoptar un lenguaje ya existente como FORTRAN, algo que propuso el Departamento de Defensa, el resto de involucrados propuso hacer un nuevo lenguaje que sirviera para toda clase de máquinas, presentes y futuras. Además, era necesario que este se pudiera adaptar a distintos entornos o actividades. Y así fue.

Entre sus creadores destaca Grace Hopper, científica computacional y oficial de la Marina de Estados Unidos. Inventora del lenguaje de programación FLOW-MATIC, también se la considera la madre de COBOL.

El éxito de COBOL fue tal que, desde su nacimiento en 1959, fue haciéndose popular hasta ponerse en el primer puesto allá por 1970. Bancos, empresas y organismos gubernamentales usaban máquinas programables con COBOL para procesar datos internos. Y, a lo largo de los años, sus especificaciones han ido evolucionando notablemente. A la versión original, COBOL 60, le sucedieron varias versiones o revisiones diferenciadas por el año de creación de cada revisión. Así, tenemos COBOL 68, COBOL 74, COBOL 85 y los más recientes, COBOL 2002 y COBOL 2014.

Revisiones necesarias si tenemos en cuenta que este lenguaje de programación tenía el propósito de convertirse en un estándar en la industria de la computación. Y que su uso se enfoca en entornos y propósitos de gran envergadura como censos, padrones, información bancaria de relevancia, subsidios y ayudas y un largo etcétera de herramientas que gestionan enormes cantidades de datos.

¿Por qué COBOL sigue siendo relevante tras más de 60 años?

Ahora que sabemos de dónde viene COBOL, se entiende en parte por qué este lenguaje de programación tuvo éxito y siguió vivo pese a la introducción de otros lenguajes que hoy son referentes e imprescindibles para cualquier programador. Python, C, Java o JavaScript lideran las listas de lenguajes más demandados. Pero, en

determinados sectores, COBOL no solo sigue presente, sino que es, incluso, el principal lenguaje de programación.

Por un lado, se siguen programando aplicaciones con lenguaje COBOL –en paralelo a la modernización de aplicaciones COBOL previas–. Especialmente, en el caso de piezas de código empleadas en computación en la nube. Es más: empresas como Amazon proveen a sus clientes de herramientas que permiten migrar sus aplicaciones COBOL para, así, poder usarlas con sus servicios en la nube de Amazon Web Services.

En palabras de Amazon, "COBOL es la base de numerosas aplicaciones troncales empresariales. COBOL sigue siendo muy utilizado en organizaciones que utilizan mainframes, como bancos, compañías de seguros y administraciones públicas". Y pone de relevancia que "antiguas aplicaciones COBOL siguen siendo muy utilizadas por varios estados de EE.UU. para procesar las solicitudes de subsidio de desempleo".

Así, COBOL se nutre de viejas máquinas que todavía siguen usando antiguas aplicaciones en entornos delicados con aplicaciones recién programadas y la modernización de viejas aplicaciones que todavía tienen un papel importante. De ahí que se sigan escribiendo miles de líneas de código en lenguaje COBOL, según datos de IBM.

NO COPIAR

Conociendo el Lenguaje Cobol

Un programa COBOL se estructura en cuatro divisiones principales: **IDENTIFICATION DIVISION**, **ENVIRONMENT DIVISION**, **DATA DIVISION** y **PROCEDURE DIVISION**. Cada división tiene un propósito específico en la organización y ejecución de un programa COBOL.

A continuación, se detallan las partes principales de un programa COBOL:

1. IDENTIFICATION DIVISION

Esta división identifica el programa COBOL y proporciona información sobre él, como el nombre del programa, autor, fecha de creación, etc.

Es la primera división que aparece en un programa COBOL.

La sección más importante de esta división es PROGRAM-ID, que define el nombre único del programa.

2. ENVIRONMENT DIVISION

Esta división especifica el entorno en el que se ejecuta el programa, como el hardware y el sistema operativo.

También define los archivos que se utilizan en el programa, tanto de entrada como de salida.

Consta de dos secciones: CONFIGURATION SECTION (especifica el entorno de compilación y ejecución) e I-O-CONTROL (describe los archivos de entrada/salida).

3. DATA DIVISION

Esta división define todos los datos que se utilizan en el programa, incluyendo variables, estructuras de datos y archivos.

Se divide en varias secciones: FILE SECTION (describe los archivos), WORKING-STORAGE SECTION (define variables), LOCAL-STORAGE SECTION (variables locales a procedimientos) y LINKAGE SECTION (variables compartidas con otros programas).

4. PROCEDURE DIVISION

Esta división contiene el código ejecutable del programa, es decir, las instrucciones y la lógica para realizar las operaciones.

Es donde se escriben los algoritmos y los pasos para procesar los datos.

NO COPIAR

Comienza con la palabra clave PROCEDURE DIVISION y termina con STOP RUN o EXIT PROGRAM.

En resumen, estas cuatro divisiones trabajan juntas para crear un programa COBOL completo y funcional. La estructura jerárquica de las divisiones facilita la organización, comprensión y mantenimiento del código.

No se preocupen, esto lo iremos viendo posteriormente en los ejemplos, pero es esencial reconocer como se comportan, después hablaremos de las Hojas, Palabras Reservadas y desarrollaremos nuestro primer programa Cobol.

5. Hojas de Codificación

En el contexto del lenguaje de programación COBOL, la "hoja" se refiere a una estructura de texto que contiene las líneas de código COBOL. Cada línea en una hoja COBOL representa una instrucción o parte de una instrucción. Estas hojas se dividen en áreas específicas con diferentes propósitos, como la numeración de secuencia, la identificación de la zona de edición y la zona de texto principal.

Se refiere al formato tradicional de las tarjetas perforadas que se utilizaban para ingresar código COBOL. Aunque hoy en día se utiliza software para escribir código COBOL, el formato de la hoja de codificación sigue siendo relevante para entender la estructura del lenguaje.

- **Áreas de la Hoja:**
    - **Área de Secuencia (Columnas 1-6):** Antiguamente, se utilizaba para numerar las líneas del código, permitiendo un ordenamiento y revisión. Hoy en día, esta función está obsoleta gracias a los entornos de desarrollo integrados (IDEs).
    - **Área de Identificación (Columna 7):** Se utilizaba para indicar si una línea era parte de una secuencia continua o una nueva sección.
    - **Área de Programa (Columnas 8-72):** Aquí se escribe la lógica del programa, es decir, las instrucciones COBOL en sí mismas.
        - **Área de Comentario (Columnas 73-80):** Se utilizaba para añadir comentarios al código, los cuales no eran procesados por el compilador.
- **Hojas de Codificación en la Actualidad:**

  Aunque las tarjetas perforadas ya no se utilizan, los programadores COBOL siguen trabajando con un formato similar en sus editores de texto. La estructura

NO COPIAR

de la hoja de codificación sigue siendo importante para comprender la sintaxis y la organización del código.

En resumen, la hoja de codificación COBOL es una estructura que define la forma en que se escribe el código COBOL, dividiendo el espacio en áreas con funciones específicas, aunque la numeración secuencial y la identificación de líneas se han vuelto obsoletas con el uso de IDEs modernos.

6. Palabras Reservadas

Palabra	Descripción	Ejemplo
ACCEPT	Recibe datos del usuario	ACCEPT variable.
ADD	Realiza una suma y almacena el resultado en una variable	ADD num1 TO num2 GIVING result.
CALL	Llama a otro programa o subprograma	CALL 'program-name'.
CLOSE	Cierra un archivo abierto.	CLOSE file-name.
COMPUTE	Realiza cálculos aritméticos complejos	COMPUTE result = num1 + num2 * num3.
DIVIDE	Realiza una división y puede almacenar el resultado	DIVIDE num1 BY num2 GIVING result.
DISPLAY	Muestra información en la salida estándar.	DISPLAY 'Hello, World!'.
EVALUATE	Realiza comparaciones, similar a un switch	EVALUATE RESTA-11 WHEN 0 MOVE "0" TO WS-DV WHEN 1 MOVE "1" TO WS-DV WHEN 2 MOVE "2" TO WS-DV WHEN 3 MOVE "3" TO WS-DV WHEN 4 MOVE "4" TO WS-DV WHEN 5 MOVE "5" TO WS-DV WHEN 6 MOVE "6" TO WS-DV WHEN 7 MOVE "7" TO WS-DV WHEN 8 MOVE "8" TO WS-DV WHEN 9 MOVE "9" TO WS-DV WHEN 10 MOVE "K" TO WS-DV WHEN 11 MOVE "0" TO WS-DV END-EVALUATE

NO COPIAR

EXIT	Termina la ejecución del programa o bloque actual	EXIT PROGRAM.
FIND	Busca un registro en un archivo	FIND record-name IN file-name.

IF	Inicia una estructura de control condicional	IF LK-DV NOT = WS-DV THEN
		MOVE 1 TO LK-CODRET
		ELSE
		MOVE 0 TO LK-CODRET
		END-IF

INITIALIZE	Establece el valor de un campo a su valor inicial definido.	INITIALIZE variable-name.
INSPECT	Permite modificar el contenido de una variable y realizar análisis de cadenas.	INSPECT variable REPLACING ALL 'a' BY 'b'. INSPECT variable TALLYING Campo-numerico FOR ALL "."
MOVE	Copia un valor de una variable a otra.	MOVE source TO destination.
OPEN	Abre un archivo para operaciones de entrada/salida.	OPEN INPUT file-name. OPEN I-O file-name OPEN OUTPUT file-name OPEN EXTENDE file-name
PERFORM	Ejecuta un procedimiento o un conjunto de instrucciones.	PERFORM paragraph-name.
READ	Lee un registro de un archivo.	READ file-name INTO variable-name. READ file-name END-READ READ file-name INVALID KEY DISPLAY "Mensaje No encontrado." NOT INVALID KEY // Accion a realizar si existe Registro END-READ. READ file-name AT END MOVE 'Y' TO fin-de-archivo NOT AT END

		// Accion a realizar si existe Registro END-READ
REWRITE	Reescribe un registro en un archivo	REWRITE record-name.
SEARCH	Busca a través de un grupo de registros o elementos	SEARCH table-name WHEN condition.
SET	Asigna valores a variables o punteros	SET variable TO value.
STOP	Finaliza la ejecución del programa	STOP RUN.
SUBTRACT	Resta un valor de otro y puede almacenar el resultado	SUBTRACT num1 FROM num2 GIVING result.
THROUGH	Especifica el flujo de un bucle	PERFORM VARYING index FROM 1 BY 1 UNTIL index > 10.
WRITE	Escribe un registro en un archivo	WRITE record-name.
STATUS	Proporciona el estado de una operación de archivo	IF file-status NOT = 0 THEN DISPLAY 'Error'.
LINKAGE	Define variables que se pasan a un programa llamado	LINKAGE SECTION. 01 PAR-INPUT. 05 RUT-CLIENTE PIC 9(10).
USAGE	Define el tipo de uso de una variable (como DISPLAY, COMPUTE, etc.)	88 level is USAGE DISPLAY.
EXCEPTION	Maneja condiciones excepcionales.	EXCEPTION SECTION.

NO COPIAR

7. File Status

A continuación, se muestran los Código de Status más comunes para cuando se manejan archivos.

Status	Descripción
00	Finalización exitosa de la solicitud.
02	Solicitud completada correctamente. Hay registros base adicionales con la misma clave alternativa.
04	La solicitud se completó correctamente, sin embargo, el tamaño del registro no coincide con la longitud fija del registro definida dentro del programa COBOL.
10	Fin del archivo.
14	El número de registro relativo proporcionado excede el máximo para este archivo.
21	En las escrituras secuenciales, la clave es inferior a la clave escrita previamente. O bien, en la reescritura de un registro existente, el programa de aplicación cambió la clave del registro.
22	Se intentó ESCRIBIR un nuevo registro para una clave que coincide con un registro que se encuentra actualmente en el archivo.
23	El registro solicitado (clave) no se encontró en el archivo.
24	Se requirió espacio DASD adicional para esta solicitud, pero no se pudo obtener. En archivos con formato compatible, esto indica que el área de desbordamiento independiente se ha llenado. En archivos con formato mejorado, el espacio DASD era insuficiente para expandir el archivo o este había alcanzado el límite máximo de extensiones permitido, que es de 16 extensiones por volumen DASD. Generalmente, este error requiere la reorganización del archivo, posiblemente con DELETE y DEFINE para asignar más espacio DASD. En archivos con formato mejorado, si la causa es la falta de espacio DASD, si se pueden eliminar algunos conjuntos de datos existentes del volumen DASD, es posible que se pueda reintentar la solicitud posteriormente.
34	El archivo está lleno. Consulte la descripción del código de estado 24 más arriba.
35	APERTURA incorrecta de un archivo vacío (descargado), como abrir solo para entrada.
37	Modo no válido para ABIERTO
39	Este estado de archivo generalmente implica una discrepancia entre los atributos del archivo IAM definidos y el diseño del registro en el programa COBOL. Por ejemplo, las posibles causas son que la longitud de la clave o el desplazamiento de la

NO COPIAR

	definición del archivo no coincidan con el diseño del registro COBOL. Además, la longitud máxima de registro definida para el archivo es menor que la longitud máxima posible para registros de longitud variable. COBOL requiere que la longitud del registro definido sea al menos igual al tamaño máximo teórico del registro, según el diseño. Consulte la función IAM PSEUDOLRECL para evitar esta restricción.
41	Se solicitó APERTURA para un archivo que ya había sido abierto.
42	Se emitió un CIERRE para un archivo que ya estaba cerrado.
43	Se emitió una solicitud de actualización de archivo (es decir, REWRITE o DELETE) sin una LECTURA previa para ACTUALIZAR.
44	Se especificó una longitud de registro incorrecta en la escritura. La longitud era menor que la mínima (es decir, la longitud de la clave más el desplazamiento de la clave), o bien el registro era mayor que el tamaño máximo definido para el archivo.
46	Una LECTURA falló porque la aplicación no había establecido exitosamente una posición en el archivo, o se intentó una LECTURA después de que se alcanzó el final del archivo.
47	Se emitió una solicitud de LECTURA para un archivo que no se abrió para entrada o E/S.
48	Se emitió una solicitud de ESCRITURA para un archivo que no está abierto para salida o E/S (actualización).
49	Se emitió una solicitud DELETE o REWRITE para un archivo que no se abrió para E/S (actualización).
90	Error lógico. Posible código de error lógico de IAM
92	Error lógico. Los atributos del archivo del programa COBOL, como la longitud de la clave, la posición relativa de la clave o la longitud del registro, no coinciden con el archivo IAM al que se accede.
93	Errores de contención de bloqueo de registros o control exclusivo. También podría deberse a un almacenamiento virtual insuficiente para procesar la solicitud. Lo más probable es que se deba aumentar el parámetro REGION para el trabajo. Consulte la salida IAM, IAMPRINT, de un Listcat del archivo para determinar cuánto almacenamiento se requiere para abrirlo.
94	No posicionado para una solicitud de LECTURA secuencial.
95	Información de archivo inválida o incompleta.
96	Falta la declaración DD para este archivo.

NO COPIAR

Fuentes del Manual

Todos los fuentes que se desarrollaran en el siguiente Manual estarán disponibles para bajarlos y probarlos con OpenCobolIDE, además el código es lo más básico para que pueda ser usado en diferentes ambientes, sobre todos los programas tipo Rutinas.

- https://github.com/jorgeduarteurzua/CobolEjemplosPracticos

Instalación OpenCobolIde

A continuación se deja Link para descargar OpenCobolIDE, este se encuentra en la siguiente Ruta.

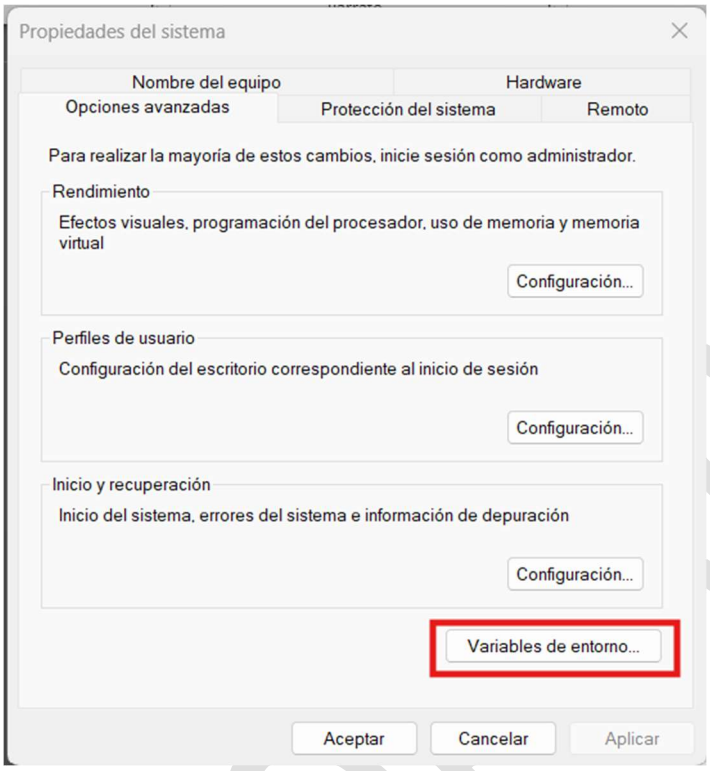
- https://drive.google.com/drive/folders/10GS2NJsKvVqZDcYe4M_2uZVKVr4E_0N5?usp=drive_link
- https://github.com/jorgeduarteurzua/CobolEjemplosPracticos

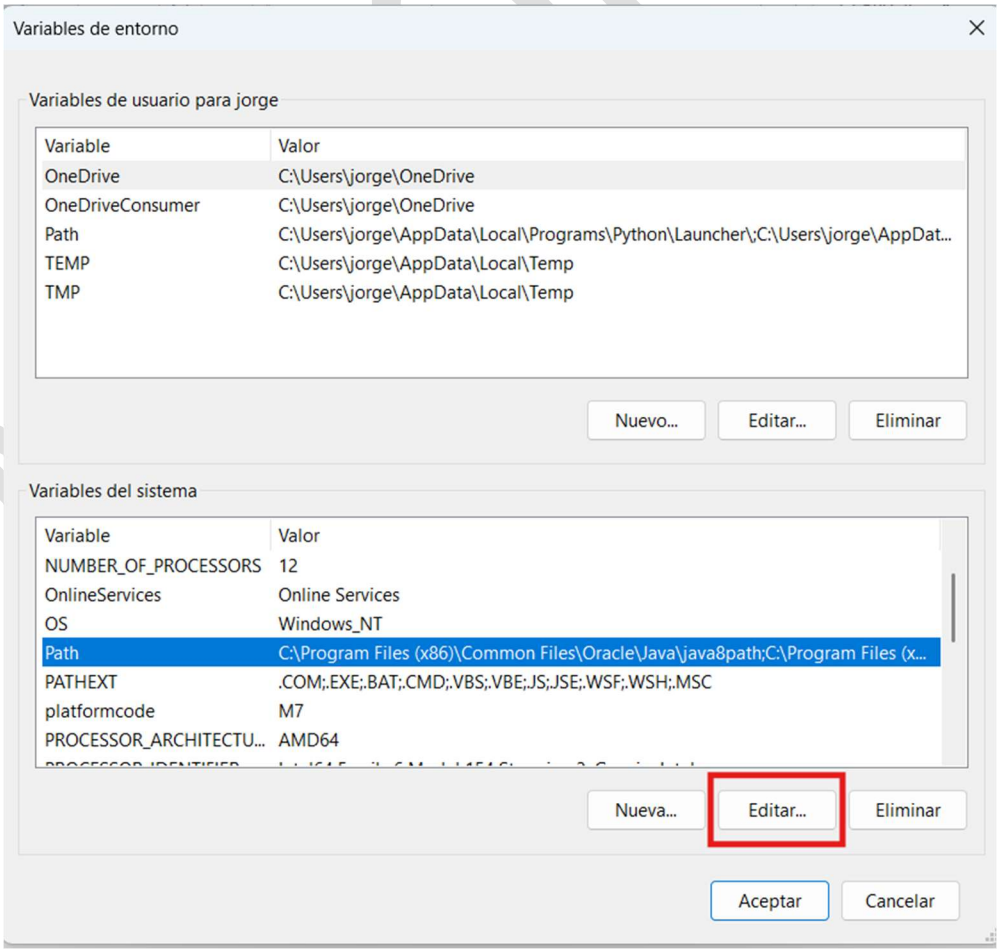
Sin embargo el link original para descarga es el siguiente, se recomienda revisar el siguiente link por si existe alguna versión nueva.

- https://launchpad.net/cobcide/+download

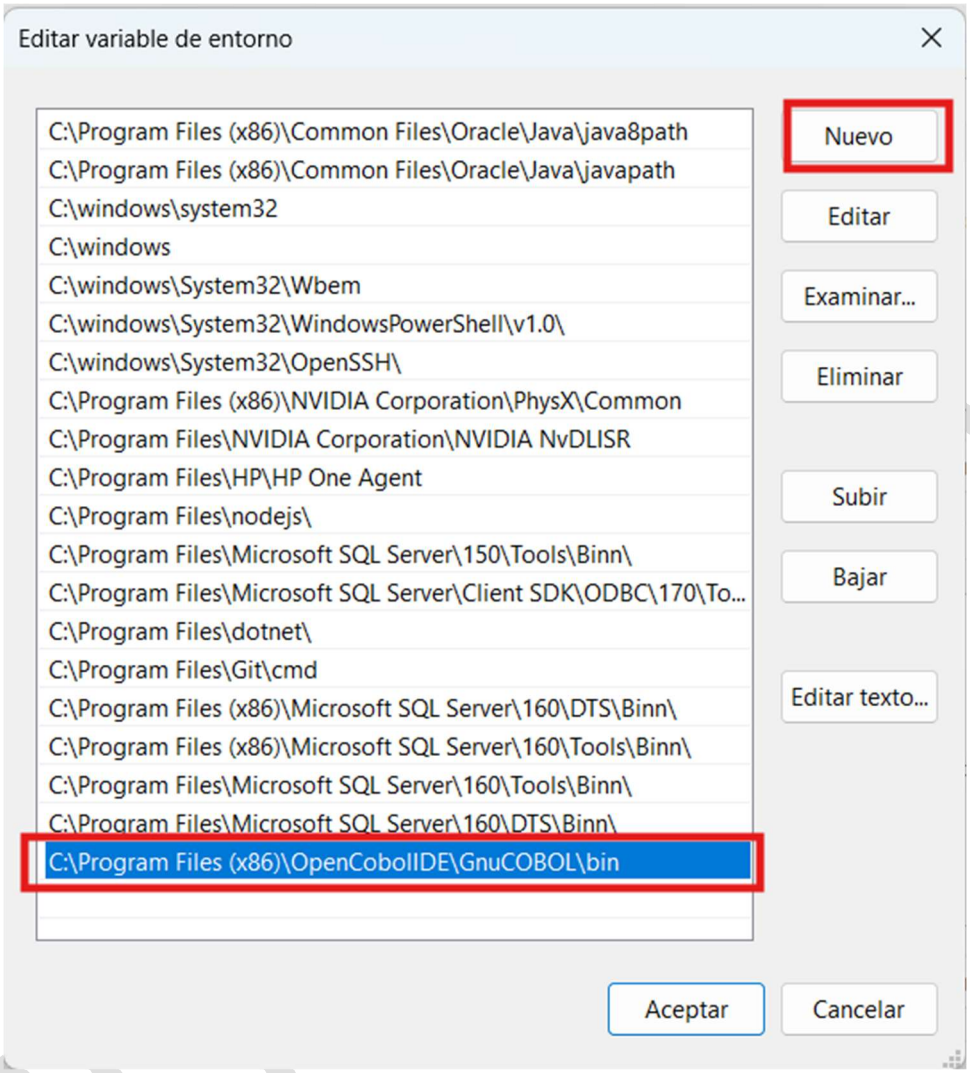
Una vez instalado se debe setear en Windows las siguiente Variable de Ambiente.

NO COPIAR

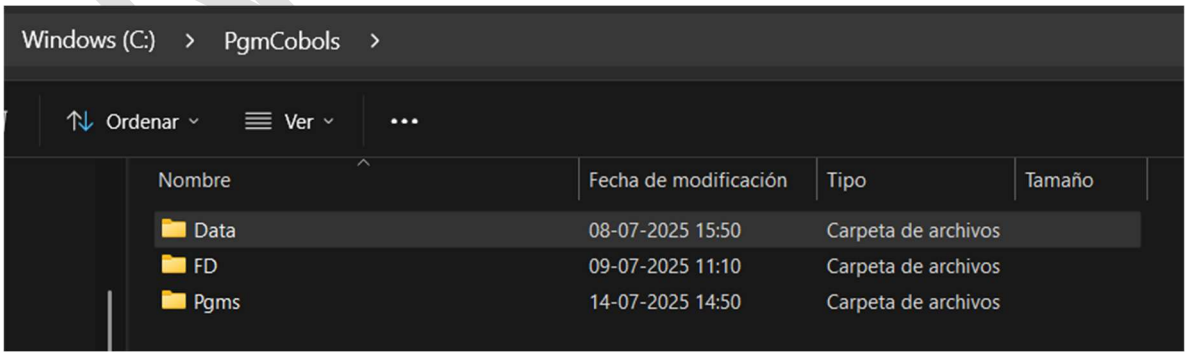

Propiedades del sistema
Nombre del equipo
Hardware
Opciones avanzadas
Protección del sistema
Remoto
Para realizar la mayoría de estos cambios, inicie sesión como administrador.
Rendimiento
Efectos visuales, programación del procesador, uso de memoria y memoria virtual
Configuración...
Perfiles de usuario
Configuración del escritorio correspondiente al inicio de sesión
Configuración...
Inicio y recuperación
Inicio del sistema, errores del sistema e información de depuración
Configuración...
Variables de entorno...
Aceptar
Cancelar
Aplicar


Variables de entorno
Variables de usuario para jorge
Variable
Valor
OneDrive
C:\Users\jorge\OneDrive
OneDriveConsumer
C:\Users\jorge\OneDrive
Path
C:\Users\jorge\AppData\Local\Programs\Python\Launcher\;C:\Users\jorge\AppDat...
TEMP
C:\Users\jorge\AppData\Local\Temp
TMP
C:\Users\jorge\AppData\Local\Temp
Nuevo...
Editar...
Eliminar
Variables del sistema
Variable
Valor
NUMBER_OF_PROCESSORS
12
OnlineServices
Online Services
OS
Windows_NT
Path
C:\Program Files (x86)\Common Files\Oracle\Java\java8path;C:\Program Files (x...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
platformcode
M7
PROCESSOR_ARCHITECTU...
AMD64
Nueva...
Editar...
Eliminar
Aceptar
Cancelar

Se debe agregar la Ruta donde quedó instalado OpenCobolIde.



La estructura que usaremos es la siguiente:



Data: Contendrá los Archivos que usaremos en este manual

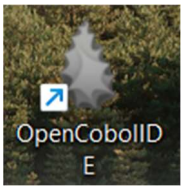
FD : Contendrá los File Description, estructura de los Archivos que usaremos los cuales pueden ser **.cbl** ó .**cpy**.

Pgms: Contendrá todos los programas Cobol que desarrollaremos en este Manual.

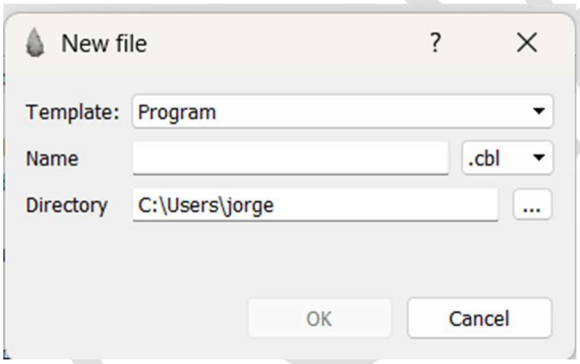
NO COPIAR

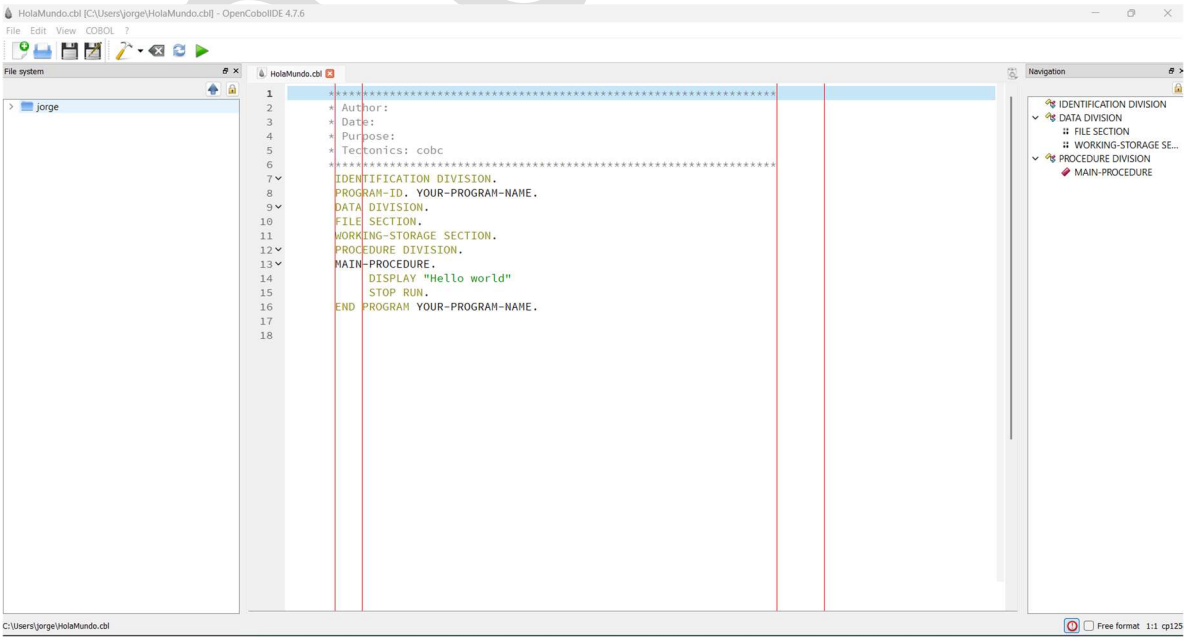
Primer Programa Cobol (HolaMundo)

Abrimos OpenCobolIde

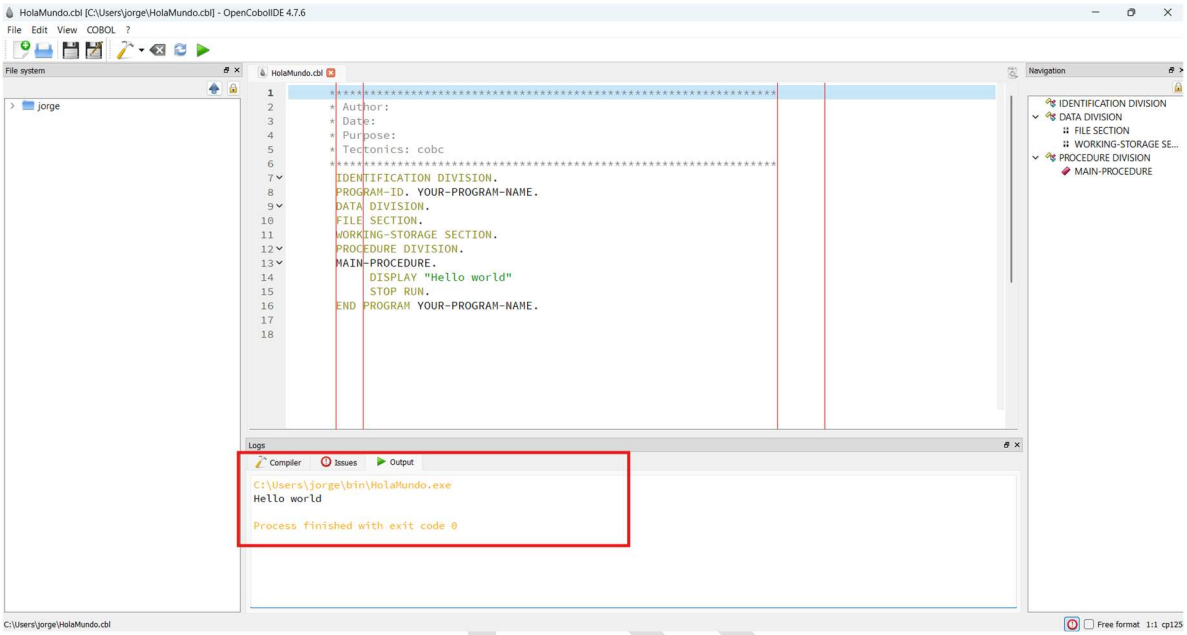


Agregamos nuevo Archivo, y creamos el Programa **HolaMundo**.





Para ejecutar presionamos la Flecha Verde, y si todo sale bien, debería salir “Hello World por consola”.



No co

Programas Útiles

A continuación entregaremos varios programas que son útiles en el día a día, sin embargo muchos pueden ser resueltos con Funciones o nuevas implementaciones del Lenguaje, sin embargo a veces estas herramientas no se encuentran disponibles y es necesario desarrollarlas e implementarlas, espero sean de utilidad y manos a la obra.

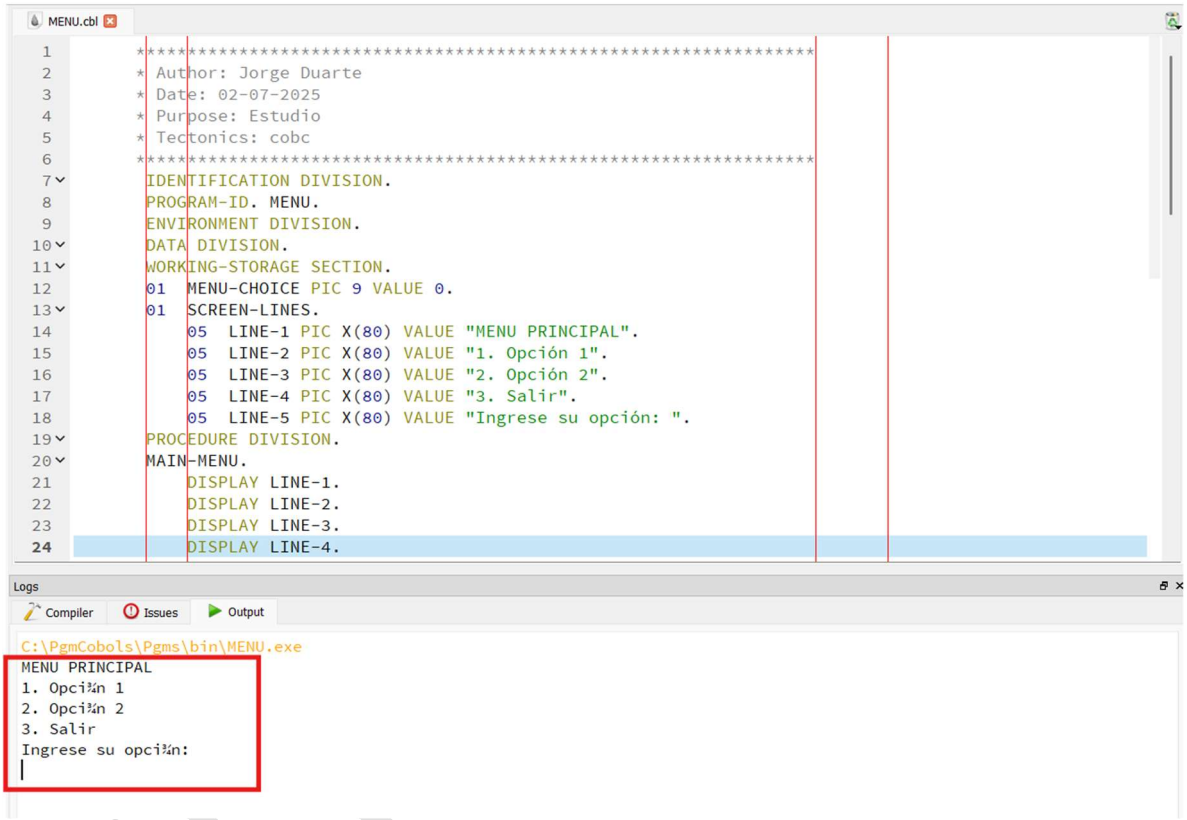
Menú Básico (MENU.cbl)

El siguiente programa tiene como finalidad mostrar un pequeño menú, sólo con las instrucciones Display y Accept, no se utiliza SCREEN SECTION para mostrar distintas formas de realizar menús.

```
******************************************************************
* Author: Jorge Duarte
* Date: 02-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. MENU.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 01  MENU-CHOICE PIC 9 VALUE 0.
 01  SCREEN-LINES.
     05  LINE-1 PIC X(80) VALUE "MENU PRINCIPAL".
     05  LINE-2 PIC X(80) VALUE "1. Opción 1".
     05  LINE-3 PIC X(80) VALUE "2. Opción 2".
     05  LINE-4 PIC X(80) VALUE "3. Salir".
     05  LINE-5 PIC X(80) VALUE "Ingrese su opción: ".
 PROCEDURE DIVISION.
 MAIN-MENU.
     DISPLAY LINE-1.
     DISPLAY LINE-2.
     DISPLAY LINE-3.
     DISPLAY LINE-4.
     DISPLAY LINE-5.
     ACCEPT MENU-CHOICE.
     EVALUATE MENU-CHOICE
       WHEN 1
         PERFORM OPTION-1
       WHEN 2
         PERFORM OPTION-2
       WHEN 3
         PERFORM EXIT-PROGRAM
       WHEN OTHER
         DISPLAY "Opción inválida. Intente de nuevo."
         GO TO MAIN-MENU
     END-EVALUATE.

 OPTION-1.
     DISPLAY "Has elegido la Opción 1".
```

NO COPIAR

```
            GO TO MAIN-MENU.

        OPTION-2.
            DISPLAY "Has elegido la Opción 2".
            GO TO MAIN-MENU.

        EXIT-PROGRAM.
            DISPLAY "Saliendo del programa.".
        STOP RUN.
```



Cálcula Digito Verificador Rut (CalculaDVRut.CBL)

El siguiente programa Calcula el Digito Verificador de un Rut, esta rutina utiliza el Módulo 10, por ejemplo en Chile se usa esta rutina para Calcular el Rut de una persona que se encuentra inscrita en el Registro Civil de Chile.

```
******************************************************************
* Author: JORGE DUARTE
* Date: 07-07-2025
* Purpose: Enseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. CalculaDVRut.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 WS-RUT     PIC 9(10).
     05 WS-DV      PIC X.
     05 ENTERO     PIC 9(2).
     05 RESTO      PIC 9(2).
     05 RESTA-11   PIC 9(3).
     05 I-MULT     PIC 9.
     05 SUMA-MULT  PIC 9(5).

 LINKAGE SECTION.
 01 PARAMETRO-ENTRADA.
     05 LK-RUT    PIC 9(10).
 01 PARAMETRO-SALIDA.
     05 LK-DV     PIC X.

 PROCEDURE DIVISION USING PARAMETRO-ENTRADA
                          PARAMETRO-SALIDA.
 MAIN-PROCEDURE.

     MOVE LK-RUT      TO WS-RUT
     MOVE 2           TO I-MULT
     MOVE 0           TO SUMA-MULT
     PERFORM UNTIL WS-RUT = 0
        DIVIDE WS-RUT BY 10 GIVING WS-RUT
               REMAINDER RESTO
        IF I-MULT > 7
           MOVE 2   TO I-MULT
        END-IF
        COMPUTE SUMA-MULT = SUMA-MULT + (RESTO * I-MULT)
        ADD 1       TO I-MULT
     END-PERFORM

     DIVIDE SUMA-MULT BY 11 GIVING ENTERO
                            REMAINDER RESTO
     COMPUTE RESTA-11 = 11 - RESTO
     EVALUATE RESTA-11
       WHEN 0   MOVE "0"   TO WS-DV
       WHEN 1   MOVE "1"   TO WS-DV
       WHEN 2   MOVE "2"   TO WS-DV
       WHEN 3   MOVE "3"   TO WS-DV
       WHEN 4   MOVE "4"   TO WS-DV
       WHEN 5   MOVE "5"   TO WS-DV
       WHEN 6   MOVE "6"   TO WS-DV
       WHEN 7   MOVE "7"   TO WS-DV
```

NO COPIAR

```
          WHEN 8   MOVE "8"   TO WS-DV
          WHEN 9   MOVE "9"   TO WS-DV
          WHEN 10  MOVE "K"   TO WS-DV
          WHEN 11  MOVE "0"   TO WS-DV
        END-EVALUATE

        MOVE WS-DV           TO LK-DV

        GOBACK.


    END PROGRAM CalculaDVRut.
```

NO COPIAR

Valida Rut (ValidaRut.CBL)

La siguiente Rutina, recibe como parámetro un Rut y un Dígito Verificador y devuelve una respuesta con valores 0=Si es correcto el Rut, 1=Si es incorrecto el Rut.

```
******************************************************************
    * Author: JORGE DUARTE
    * Date: 07-07-2025
    * Purpose: Enseñanza
    * Tectonics: cobc
    ******************************************************************
     IDENTIFICATION DIVISION.
     PROGRAM-ID. ValidaRut.
     DATA DIVISION.
     FILE SECTION.
     WORKING-STORAGE SECTION.
     01 VARIABLES-DE-TRABAJO.
         05 WS-RUT     PIC 9(10).
         05 WS-DV      PIC X.
         05 ENTERO     PIC 9(2).
         05 RESTO      PIC 9(2).
         05 RESTA-11   PIC 9(3).
         05 I-MULT     PIC 9.
         05 SUMA-MULT  PIC 9(5).

     LINKAGE SECTION.
     01 PARAMETRO-ENTRADA.
         05 LK-RUT    PIC 9(10).
         05 LK-DV     PIC X.
     01 PARAMETRO-SALIDA.
         05 LK-CODRET PIC 9.
    *       0 = OK
    *       1 = NO OK
     PROCEDURE DIVISION USING PARAMETRO-ENTRADA
                              PARAMETRO-SALIDA.
     MAIN-PROCEDURE.

         MOVE LK-RUT      TO WS-RUT
         MOVE 2           TO I-MULT
         MOVE 0           TO SUMA-MULT
         PERFORM UNTIL WS-RUT = 0
            DIVIDE WS-RUT BY 10 GIVING WS-RUT
                   REMAINDER RESTO
            IF I-MULT > 7
               MOVE 2   TO I-MULT
            END-IF
            COMPUTE SUMA-MULT = SUMA-MULT + (RESTO * I-MULT)
            ADD 1       TO I-MULT
         END-PERFORM

         DIVIDE SUMA-MULT BY 11 GIVING ENTERO
                              REMAINDER RESTO
         COMPUTE RESTA-11 = 11 - RESTO
         EVALUATE RESTA-11
           WHEN 0   MOVE "0"   TO WS-DV
           WHEN 1   MOVE "1"   TO WS-DV
           WHEN 2   MOVE "2"   TO WS-DV
           WHEN 3   MOVE "3"   TO WS-DV
           WHEN 4   MOVE "4"   TO WS-DV
           WHEN 5   MOVE "5"   TO WS-DV
           WHEN 6   MOVE "6"   TO WS-DV
           WHEN 7   MOVE "7"   TO WS-DV
```

NO COPIAR

```
      WHEN 8   MOVE "8"   TO WS-DV
      WHEN 9   MOVE "9"   TO WS-DV
      WHEN 10  MOVE "K"   TO WS-DV
      WHEN 11  MOVE "0"   TO WS-DV
    END-EVALUATE

    IF LK-DV NOT = WS-DV THEN
        MOVE 1     TO LK-CODRET
    ELSE
        MOVE 0     TO LK-CODRET
    END-IF

    GOBACK.


END PROGRAM ValidaRut.
```

NO COPIAR

Probar ValidaRut y CalulaDVRut (PruebaValidaRut.CBL)

El siguiente programa muestra la ejecución y como probar las 2 rutinas declaradas anteriormente, CalcularDVRut y ValidaRut.

```
******************************************************************
* Author: JORGE DUARTE
* Date: 07-07-2025
* Purpose: Enseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. PruebaValidaRut.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 PAR-INPUT-VALIDA-RUT.
     10 WS-RUT PIC 9(10) VALUE 12961577.
     10 WS-DV  PIC X    VALUE '8'.
 01 PAR-OUTPUT-VALIDA-RUT.
     10 WS-CODRET PIC 9.

 01 PAR-INPUT-CALCULA-DV.
     10 WS-RUT-DV PIC 9(10) VALUE 12961577.
 01 PAR-OUTPUT-CALCULA-DV.
     10 WS-CODRET-DV PIC X.

 PROCEDURE DIVISION.
 MAIN-PROCEDURE.

     MOVE 0   TO WS-CODRET
     CALL "ValidaRut" USING PAR-INPUT-VALIDA-RUT
                            PAR-OUTPUT-VALIDA-RUT

     DISPLAY "Respuesta ValidaRut Rut : " WS-RUT " - " WS-DV
             " Retorno Rutina : "  WS-CODRET


      CALL "CalculaDVRut" USING PAR-INPUT-CALCULA-DV
                                PAR-OUTPUT-CALCULA-DV

     DISPLAY "Respuesta CalculaDVRut Rut : " WS-RUT-DV
             " Digito Verificador Calculado : "  WS-CODRET-DV

      STOP RUN.
 END PROGRAM PruebaValidaRut.
```

NO COPIAR

PruebaValidaRut.cbl

```
25
26          MOVE 0   TO WS-CODRET
27          CALL "ValidaRut" USING PAR-INPUT-VALIDA-RUT
28                                 PAR-OUTPUT-VALIDA-RUT
29
30          DISPLAY "Respuesta ValidaRut Rut : " WS-RUT " - " WS-DV
31                  " Retorno Rutina : "  WS-CODRET
32
33
34           CALL "CalculaDVRut" USING PAR-INPUT-CALCULA-DV
35                                     PAR-OUTPUT-CALCULA-DV
36
37          DISPLAY "Respuesta CalculaDVRut Rut : " WS-RUT-DV
38                  " Digito Verificador Calculado : "  WS-CODRET-DV
39
40           STOP RUN.
41        END PROGRAM PruebaValidaRut.
42
```

Logs

Compiler | Issues | Output

```
C:\PgmCobols\Pgms\bin\PruebaValidaRut.exe
Respuesta ValidaRut Rut : 0012961577 - 8 Retorno Rutina : 1
Respuesta CalculaDVRut Rut : 0012961577 Digito Verificador Calculado : K

Process finished with exit code 0
```

NO CO

Calcular Largo String Máx 500 Caracteres (LargoString.CBL)

```
******************************************************************
* Author: JORGE DUARTE
* Date: 08-07-2025
* Purpose: Eseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. LargoString.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 LARGO-STR PIC 9(03).

 LINKAGE SECTION.
 01 INPUT-LARGOSTR500.
     05 STRING-CALCULAR500   PIC X(500).
 01 OUTPUT-LARGOSTR500.
     05 LK-LARGO-STR500      PIC 9(03).

 PROCEDURE DIVISION USING INPUT-LARGOSTR500
                          OUTPUT-LARGOSTR500.

 MAIN-PROCEDURE.

     PERFORM VARYING LARGO-STR FROM 500 BY -1 UNTIL
             LARGO-STR = 0 OR
             STRING-CALCULAR500(LARGO-STR:1) NOT = SPACES
     END-PERFORM

     MOVE LARGO-STR   TO LK-LARGO-STR500

     GOBACK
     .

 END PROGRAM LargoString.
```

NO COPIAR

Convertir String en Mayúsculas (ConvertirMayusculas.CBL)

```
******************************************************************
* Author: JORGE DUARTE
* Date: 08-07-2025
* Purpose: Eseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ConvertirMayusculas.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 I-STR     PIC 9(3).
     05 I-MAY     PIC 9(3).
     05 LARGO-STR PIC 9(03).

     05 WS-MIN PIC X(27) VALUE "abcdefghijklmnñopqrstuvwxyz".
     05 WS-MAY PIC X(27) VALUE "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ".

 LINKAGE SECTION.
 01 PAR-INPUT.
     05 STRING-CONVERTIR PIC X(500).
 01 PAR-OUTPUT.
     05 STRING-CONVERTIDO PIC X(500).

 PROCEDURE DIVISION USING PAR-INPUT
                          PAR-OUTPUT.

 MAIN-PROCEDURE.

      CALL "LargoString" using PAR-INPUT LARGO-STR

      PERFORM VARYING I-STR FROM 1 BY 1 UNTIL I-STR > LARGO-STR
          MOVE STRING-CONVERTIR (I-STR:1) TO
               STRING-CONVERTIDO(I-STR:1)
          IF STRING-CONVERTIDO(I-STR:1) NOT = SPACES
             PERFORM VARYING I-MAY FROM 1 BY 1 UNTIL I-MAY > 27
                IF STRING-CONVERTIDO(I-STR:1) =
                   WS-MIN(I-MAY:1) THEN
                   MOVE WS-MAY(I-MAY:1) TO
                        STRING-CONVERTIDO(I-STR:1)
                   ADD 28  TO I-MAY
                END-IF
             END-PERFORM
          END-IF

      END-PERFORM

      GOBACK.

 END PROGRAM ConvertirMayusculas.
```

NO COPIAR

Convertir String en Minúsculas (ConvertirMinusculas.CBL)

```
******************************************************************
* Author: JORGE DUARTE
* Date: 08-07-2025
* Purpose: Eseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ConvertirMinusculas.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 I-STR     PIC 9(3).
     05 I-MIN     PIC 9(3).
     05 LARGO-STR PIC 9(03).

     05 WS-MIN PIC X(27) VALUE "abcdefghijklmnñopqrstuvwxyz".
     05 WS-MAY PIC X(27) VALUE "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ".

 LINKAGE SECTION.
 01 PAR-INPUT.
     05 STRING-CONVERTIR PIC X(500).
 01 PAR-OUTPUT.
     05 STRING-CONVERTIDO PIC X(500).

 PROCEDURE DIVISION USING PAR-INPUT
                          PAR-OUTPUT.

 MAIN-PROCEDURE.

      CALL "LargoString" using PAR-INPUT LARGO-STR

      PERFORM VARYING I-STR FROM 1 BY 1 UNTIL I-STR > LARGO-STR
          MOVE STRING-CONVERTIR (I-STR:1) TO
               STRING-CONVERTIDO(I-STR:1)
          IF STRING-CONVERTIDO(I-STR:1) NOT = SPACES
             PERFORM VARYING I-MIN FROM 1 BY 1 UNTIL I-MIN > 27
                IF STRING-CONVERTIDO(I-STR:1) =
                   WS-MAY(I-MIN:1) THEN
                   MOVE WS-MIN(I-MIN:1) TO
                        STRING-CONVERTIDO(I-STR:1)
                   ADD 28  TO I-MIN
                END-IF
             END-PERFORM
          END-IF

      END-PERFORM

      GOBACK.

 END PROGRAM ConvertirMinusculas.
```

NO COPIAR

Prueba Convertir en Mayúsculas/Minúsculas (PruebaMayusculasMinusculas.CBL)

```
      ******************************************************************
      * Author: JORGE DUARTE
      * Date: 07-07-2025
      * Purpose: Enseñanza
      * Tectonics: cobc
      ******************************************************************
       IDENTIFICATION DIVISION.
       PROGRAM-ID. PruebaMayusculasMinusculas.
       DATA DIVISION.
       FILE SECTION.
       WORKING-STORAGE SECTION.
       01 VARIABLES-DE-TRABAJO.
           05 TEXTO-TRANSFORMAR PIC X(40) VALUE "Prueba Convertir".

       01 PAR-INPUT-MAY.
           05 MAY-STRING-CONVERTIR PIC X(500).
       01 PAR-OUTPUT-MAY.
           05 MAY-STRING-CONVERTIDO PIC X(500).

       01 PAR-INPUT-MIN.
           05 MIN-STRING-CONVERTIR PIC X(500).
       01 PAR-OUTPUT-MIN.
           05 MIN-STRING-CONVERTIDO PIC X(500).

       PROCEDURE DIVISION.
       MAIN-PROCEDURE.

           MOVE TEXTO-TRANSFORMAR TO MAY-STRING-CONVERTIR
                                     MIN-STRING-CONVERTIR

           CALL "ConvertirMayusculas" USING PAR-INPUT-MAY
                                            PAR-OUTPUT-MAY

           CALL "ConvertirMinusculas" USING PAR-INPUT-MIN
                                            PAR-OUTPUT-MIN


           DISPLAY "TEXTO A CONVERTIR : " TEXTO-TRANSFORMAR
           DISPLAY "TEXTO EN MAYUSCULAS : " MAY-STRING-CONVERTIDO
           DISPLAY "TEXTO EN MINUSCULAS : " MIN-STRING-CONVERTIDO

            STOP RUN.
       END PROGRAM PruebaMayusculasMinusculas.
```

NO COPIAR

PruebaMayusculasMinusculas.cbl

```
25        PROCEDURE DIVISION.
26        MAIN-PROCEDURE.
27
28            MOVE TEXTO-TRANSFORMAR TO MAY-STRING-CONVERTIR
29                                      MIN-STRING-CONVERTIR
30
31            CALL "ConvertirMayusculas" USING PAR-INPUT-MAY
32                                             PAR-OUTPUT-MAY
33
34            CALL "ConvertirMinusculas" USING PAR-INPUT-MIN
35                                             PAR-OUTPUT-MIN
36
37
38            DISPLAY "TEXTO A CONVERTIR : " TEXTO-TRANSFORMAR
39            DISPLAY "TEXTO EN MAYUSCULAS : " MAY-STRING-CONVERTIDO
40            DISPLAY "TEXTO EN MINUSCULAS : " MIN-STRING-CONVERTIDO
41
42             STOP RUN.
43        END PROGRAM PruebaMayusculasMinusculas.
44
```

Logs

Compiler | Issues | Output

```
C:\PgmCobols\Pgms\bin\PruebaMayusculasMinusculas.exe
TEXTO A CONVERTIR : Prueba Convertir
TEXTO EN MAYUSCULAS : PRUEBA CONVERTIR
TEXTO EN MINUSCULAS : prueba convertir

Process finished with exit code 0
```

NO CO

Elimina los Blancos Inciales y Finales (Trim.CBL)

```
******************************************************************
* Author: Jorge Duarte
* Date: 14-07-2025
* Purpose: Enseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. Trim.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 STR-INICIO   PIC 9(3).

 01 PAR-INPUT-LARGO-STRING.
     05 STRING-CALULAR   PIC X(500).
 01 PAR-OUTPUT-LARGO-STRING.
     05 LK-LARGO-STR     PIC 9(03).

 LINKAGE SECTION.
 01 PAR-INPUT-TRIM.
     05 TEXTO-INICIAL PIC X(500).
 01 PAR-OUTPUT-TRIM.
     05 NEW-TEXTO     PIC X(500).
     05 LARGO-NEW-TEXTO PIC 9(03).

 PROCEDURE DIVISION USING PAR-INPUT-TRIM
                          PAR-OUTPUT-TRIM.

 MAIN-PROCEDURE.

     PERFORM VARYING
     STR-INICIO FROM 1 BY 1 UNTIL STR-INICIO > 500
             OR TEXTO-INICIAL(STR-INICIO:1) <> " "
     END-PERFORM

     MOVE TEXTO-INICIAL(STR-INICIO:) TO NEW-TEXTO
                                        STRING-CALULAR
     MOVE ZEROES                     TO LK-LARGO-STR
     CALL "LargoString" USING PAR-INPUT-LARGO-STRING
                              PAR-OUTPUT-LARGO-STRING

     MOVE LK-LARGO-STR               TO LARGO-NEW-TEXTO


     GOBACK.
 END PROGRAM Trim.
```

NO COPIAR

Convierte String en SnakeCase (SnakeCase.CBL)

```
      ******************************************************************
      * Author: Jorge Duarte
      * Date: 14-07-2025
      * Purpose: Enseñanza
      * Tectonics: cobc
      ******************************************************************
       IDENTIFICATION DIVISION.
       PROGRAM-ID. SnakeCase.
       DATA DIVISION.
       FILE SECTION.
       WORKING-STORAGE SECTION.
       01 PAR-INPUT-MIN.
           05 MIN-STRING-CONVERTIR PIC X(500).
       01 PAR-OUTPUT-MIN.
           05 MIN-STRING-CONVERTIDO PIC X(500).

       LINKAGE SECTION.
       01 PAR-INP-SNAKE.
           05 SNAKE-TEXTO-CONVERTIR PIC X(500).
       01 PAR-OUT-SNAKE.
           05 SNAKE-TEXTO-CONVERTIDO PIC X(500).

       PROCEDURE DIVISION USING PAR-INP-SNAKE
                                PAR-OUT-SNAKE.
       MAIN-PROCEDURE.

            MOVE SNAKE-TEXTO-CONVERTIR TO MIN-STRING-CONVERTIR
            MOVE SPACES                TO MIN-STRING-CONVERTIDO
            CALL "ConvertirMinusculas" USING PAR-INPUT-MIN
                                             PAR-OUTPUT-MIN

            MOVE MIN-STRING-CONVERTIDO TO SNAKE-TEXTO-CONVERTIDO
            INSPECT SNAKE-TEXTO-CONVERTIDO
                   REPLACING ALL " " BY "_"

            GOBACK.
       END PROGRAM SnakeCase.
```

NO COPIAR

Probar Trim y SnakeCase (ProbarTrimySnake.CBL)

```
******************************************************************
* Author: Jorge Duarte
* Date: 14-07-2025
* Purpose: Enseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ProbarTrimySnake.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 WS-TEXTO PIC X(500).

 01 PAR-INPUT-TRIM.
     05 TEXTO-INICIAL PIC X(500).
 01 PAR-OUTPUT-TRIM.
     05 NEW-TEXTO     PIC X(500).
     05 LARGO-NEW-TEXTO PIC 9(03).

 01 PAR-INP-SNAKE.
     05 SNAKE-TEXTO-CONVERTIR PIC X(500).
 01 PAR-OUT-SNAKE.
     05 SNAKE-TEXTO-CONVERTIDO PIC X(500).

 PROCEDURE DIVISION.
 MAIN-PROCEDURE.
      MOVE SPACES   TO WS-TEXTO
      STRING "     Este es un nuevo "    DELIMITED SIZE
             "TEXTO que usaremos para "  DELIMITED SIZE
             "probar TRIM y SNAKE CASE." DELIMITED SIZE
                                    INTO WS-TEXTO

      MOVE WS-TEXTO                  TO TEXTO-INICIAL
                                        SNAKE-TEXTO-CONVERTIR
      MOVE ZEROES                    TO LARGO-NEW-TEXTO

      DISPLAY "TEXTO USAR [" WS-TEXTO "]"

      CALL "Trim" USING PAR-INPUT-TRIM
                        PAR-OUTPUT-TRIM

      CALL "SnakeCase" USING PAR-INP-SNAKE
                             PAR-OUT-SNAKE

      DISPLAY "TRIM"
      DISPLAY "Nuevo Texto : " NEW-TEXTO(1:LARGO-NEW-TEXTO)
              " --- Largo Nuevo Texto : " LARGO-NEW-TEXTO

      DISPLAY "SNAKECASE"
      DISPLAY "Nuevo Texto : " SNAKE-TEXTO-CONVERTIDO


      STOP RUN.
 END PROGRAM ProbarTrimySnake.
```

NO COPIAR

ProbarTrimySnake.cbl

```
37
38          DISPLAY "TEXTO USAR [" WS-TEXTO "]"
39
40          CALL "Trim" USING PAR-INPUT-TRIM
41                            PAR-OUTPUT-TRIM
42
43          CALL "SnakeCase" USING PAR-INP-SNAKE
44                                 PAR-OUT-SNAKE
45
46          DISPLAY "TRIM"
47          DISPLAY "Nuevo Texto : " NEW-TEXTO(1:LARGO-NEW-TEXTO)
48                  " --- Largo Nuevo Texto : " LARGO-NEW-TEXTO
49
50          DISPLAY "SNAKECASE"
51          DISPLAY "Nuevo Texto : " SNAKE-TEXTO-CONVERTIDO
52
53
54          STOP RUN.
55      END PROGRAM ProbarTrimySnake.
```

Logs

Compiler | Issues | Output

```
C:\PgmCobols\Pgms\bin\ProbarTrimySnake.exe
TEXTO USAR [     Este es un nuevo TEXTO que usaremos para probar TRIM y SNAKE CASE.
TRIM
Nuevo Texto : Este es un nuevo TEXTO que usaremos para probar TRIM y SNAKE CASE. --- Largo Nuevo Texto : 066
SNAKECASE
Nuevo Texto : _____este_es_un_nuevo_texto_que_usaremos_para_probar_trim_y_snake_case._____________________________

Process finished with exit code 0
```

NO CO

Validar una Fecha (ValidarFecha.CBL)

La siguiente rutina recibe una Fecha y se debe indicar el Formato el cual puede ser YYYYMMDD ó DDMMYYYY, y retorna un Valor de “S”, si la Fecha es Correcta ó “N” si es incorrecta.

```
******************************************************************
* Author: JORGE DUARTE
* Date: 07-07-2025
* Purpose: Enseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ValidarFecha.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 ANO            PIC 9(04).
     05 MES            PIC 9(02).
     05 DIA            PIC 9(02).

     05 ENTERO         PIC 9(04).
     05 ENTERO-2       PIC 9(04).
     05 ENTERO-3       PIC 9(04).

     05 RESTO          PIC 9(04).
     05 RESTO-2        PIC 9(04).
     05 RESTO-3        PIC 9(04).

     05 TABLA-DIAS-MESES PIC X(24) VALUE
                       "312831303130313130313032".
     05 ARR-DIAS REDEFINES TABLA-DIAS-MESES
                 OCCURS 12 TIMES.
        10 DIA-MES     PIC 9(02).

 LINKAGE SECTION.
 01 PARAMETROS-ENTRADA.
     05 FECHA-YYYYMMDD PIC X(08).
     05 FECHA-FORMATO  PIC X.
*       1 : YYYYMMDD
*       2 : DDMMYYYY

 01 PARAMETRO-SALIDA.
     05 FECHA-VALIDA   PIC X.
*       S : FECHA ES VALIDATE
*       N : FECHA ES INVALIDA

 PROCEDURE DIVISION USING PARAMETROS-ENTRADA
                          PARAMETRO-SALIDA.
 MAIN-PROCEDURE.

     MOVE "S"            TO FECHA-VALIDA
     EVALUATE FECHA-FORMATO
       WHEN "1"
             MOVE FECHA-YYYYMMDD(1:4) TO ANO
             MOVE FECHA-YYYYMMDD(5:2) TO MES
             MOVE FECHA-YYYYMMDD(7:2) TO DIA
       WHEN "2"
             MOVE FECHA-YYYYMMDD(5:4) TO ANO
             MOVE FECHA-YYYYMMDD(3:2) TO MES
```

NO COPIAR

```
            MOVE FECHA-YYYYMMDD(1:2) TO DIA
      WHEN OTHER
            MOVE "N"         TO FECHA-VALIDA
    END-EVALUATE
    IF FECHA-VALIDA NOT = "N" THEN
        PERFORM VALIDAR-FECHA
    END-IF


    GOBACK.

VALIDAR-FECHA.

    IF MES NOT = 2 THEN
       IF MES < 1 OR > 12 THEN
          MOVE "N"   TO FECHA-VALIDA
       ELSE
           IF ANO < 1 THEN
              MOVE "N"   TO FECHA-VALIDA
           ELSE
               IF DIA < 1 OR > DIA-MES(MES) THEN
                  MOVE "N" TO FECHA-VALIDA
               END-IF
           END-IF
       END-IF
    ELSE
       PERFORM BISIESTO

*   VERIFICAMOS SI EL MES DEL AÑO ES BISIESTO
       IF (RESTO = 0 AND RESTO-2 <> 0) OR (RESTO-3 = 0) THEN
          MOVE 29    TO DIA-MES(MES)
       ELSE
          MOVE 28    TO DIA-MES(MES)
       END-IF
       IF DIA < 1 OR > DIA-MES(MES) THEN
          MOVE "N" TO FECHA-VALIDA
       END-IF
    END-IF
    .

BISIESTO.
    DIVIDE ANO BY 4 GIVING ENTERO
               REMAINDER RESTO
    DIVIDE ANO BY 100 GIVING ENTERO-2
               REMAINDER RESTO-2
    DIVIDE ANO BY 400 GIVING ENTERO-3
               REMAINDER RESTO-3
    .
END PROGRAM ValidarFecha
```

NO COPIAR

Sumar días a una Fecha (SumarDiasFecha.CBL)

La siguiente rutina, suma o resta días a una fecha, el formato de la Fecha de Entrada es YYYYMMDD, también utilizamos la rutina anterior para valida que la fecha en formato YYYYMMDD sea valida.

```
******************************************************************
* Author: JORGE DUARTE
* Date: 07-07-2025
* Purpose: Enseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. SumarDiasFecha.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.

     05 ANO-2          PIC 9(04).
     05 MES-2          PIC 9(02).
     05 DIA-2          PIC 9(02).

     05 ENTERO         PIC 9(04).
     05 ENTERO-2       PIC 9(04).
     05 ENTERO-3       PIC 9(04).

     05 RESTO          PIC 9(04).
     05 RESTO-2        PIC 9(04).
     05 RESTO-3        PIC 9(04).

     05 I              PIC S9(03).

     05 TABLA-DIAS-MESES PIC X(24) VALUE
                       "312831303130313130313031".
     05 ARR-DIAS REDEFINES TABLA-DIAS-MESES
                OCCURS 12 TIMES.
        10 DIA-MES     PIC 9(02).

 01 PARAMETROS-ENTRADA-VALFEC.
     05 VALFEC-FECHA-YYYYMMDD PIC X(08).
     05 VALFEC-FECHA-FORMATO  PIC X.
*       1 : YYYYMMDD
*       2 : DDMMYYYY

 01 PARAMETRO-SALIDA-VALFEC.
     05 VALFEC-FECHA-VALIDA   PIC X.
*       S : FECHA ES VALIDATE
*       N : FECHA ES INVALIDA

 LINKAGE SECTION.
 01 PARAMETROS-ENTRADA.
     05 FECHA-YYYYMMDD PIC X(08).
     05 SUM-RES-DIAS   PIC S9(03).
 01 PARAMETRO-SALIDA.
     05 FECHA-VALIDA         PIC X(01).
     05 NUEVA-FECHA-YYYYMMDD PIC X(08).

 PROCEDURE DIVISION USING PARAMETROS-ENTRADA
                          PARAMETRO-SALIDA.
 MAIN-PROCEDURE.
```

NO COPIAR

```
       MOVE "00000000"   TO NUEVA-FECHA-YYYYMMDD

       MOVE FECHA-YYYYMMDD(1:4) TO ANO-2
       MOVE FECHA-YYYYMMDD(5:2) TO MES-2
       MOVE FECHA-YYYYMMDD(7:2) TO DIA-2

       MOVE FECHA-YYYYMMDD TO VALFEC-FECHA-YYYYMMDD
       MOVE "1"            TO VALFEC-FECHA-FORMATO
       MOVE " "            TO VALFEC-FECHA-VALIDA
       CALL "ValidarFecha" USING PARAMETROS-ENTRADA-VALFEC
                                 PARAMETRO-SALIDA-VALFEC
       IF VALFEC-FECHA-VALIDA = "S" THEN
          IF SUM-RES-DIAS >= 0 THEN
             PERFORM SUMAR-DIAS
          ELSE
             PERFORM RESTAR-DIAS
          END-IF

       END-IF
       MOVE VALFEC-FECHA-VALIDA TO FECHA-VALIDA

       MOVE ANO-2    TO NUEVA-FECHA-YYYYMMDD(1:4)
       MOVE MES-2    TO NUEVA-FECHA-YYYYMMDD(5:2)
       MOVE DIA-2    TO NUEVA-FECHA-YYYYMMDD(7:2)

       GOBACK.

    SUMAR-DIAS.

       PERFORM VARYING I FROM 1 BY 1 UNTIL I > SUM-RES-DIAS
           ADD 1        TO DIA-2
           IF MES-2 NOT = 2 THEN
               IF DIA-2 > DIA-MES(MES-2) THEN
                   MOVE 1 TO DIA-2
                   ADD 1 TO MES-2
                   IF MES-2 > 12 THEN
                       MOVE 1   TO MES-2
                       ADD  1   TO ANO-2
                   END-IF
               END-IF
            ELSE
              PERFORM BISIESTO

    *     VERIFICAMOS SI EL MES DEL AÑO ES BISIESTO
              IF (RESTO = 0 AND RESTO-2 <> 0) OR (RESTO-3 = 0) THEN
                 MOVE 29    TO DIA-MES(MES-2)
              ELSE
                 MOVE 28    TO DIA-MES(MES-2)
              END-IF
              IF DIA-2 > DIA-MES(MES-2) THEN
                   ADD 1 TO MES-2
                   MOVE 1  TO DIA-2
                   IF MES-2 > 12 THEN
                       MOVE 1   TO MES-2
                       ADD  1   TO ANO-2
                   END-IF
                   MOVE 1       TO DIA-2
              END-IF
            END-IF

       END-PERFORM

       MOVE ANO-2    TO NUEVA-FECHA-YYYYMMDD(1:4)
```

NO COPIAR

```
    MOVE MES-2    TO NUEVA-FECHA-YYYYMMDD(5:2)
    MOVE DIA-2    TO NUEVA-FECHA-YYYYMMDD(7:2)
    .
RESTAR-DIAS.
    MULTIPLY SUM-RES-DIAS BY -1 GIVING SUM-RES-DIAS

     PERFORM VARYING I FROM 1 BY 1 UNTIL I > SUM-RES-DIAS
        SUBTRACT 1      FROM DIA-2
        IF DIA-2 < 1 THEN
            SUBTRACT 1 FROM MES-2
            IF MES-2 < 1 THEN
                SUBTRACT 1 FROM ANO-2
                MOVE 12      TO MES-2
                MOVE DIA-MES(MES-2) TO DIA-2
            ELSE
                IF MES-2 NOT = 2 THEN
                    MOVE DIA-MES(MES-2) TO DIA-2
                ELSE
                    PERFORM BISIESTO

*    VERIFICAMOS SI EL MES DEL AÑO ES BISIESTO
                   IF (RESTO = 0 AND RESTO-2 <> 0) OR
                      (RESTO-3 = 0) THEN
                      MOVE 29    TO DIA-2
                   ELSE
                      MOVE 28    TO DIA-2
                   END-IF

                END-IF
            END-IF
        END-IF

    END-PERFORM
    .
BISIESTO.
    DIVIDE ANO-2 BY 4 GIVING ENTERO
                REMAINDER RESTO
    DIVIDE ANO-2 BY 100 GIVING ENTERO-2
                REMAINDER RESTO-2
    DIVIDE ANO-2 BY 400 GIVING ENTERO-3
                REMAINDER RESTO-3
    .

END PROGRAM SumarDiasFecha.
```

NO COPIAR

Probar Sumar días a una Fecha (ProbarSumarDiasFecha.CBL)

```
******************************************************************
* Author:
* Date:
* Purpose:
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ProbarSumarDiasFecha.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 FECHA-INICIO PIC X(08) VALUE "20250714".

 01 SUMARDIAS-ENTRADA.
     05 FECHA-YYYYMMDD PIC X(08).
     05 SUM-RES-DIAS   PIC S9(03).
 01 SUMARDIAS-SALIDA.
     05 FECHA-VALIDA         PIC X(01).
     05 NUEVA-FECHA-YYYYMMDD PIC X(08).

 PROCEDURE DIVISION.
 MAIN-PROCEDURE.

     MOVE FECHA-INICIO     TO FECHA-YYYYMMDD

     MOVE ZEROES           TO SUM-RES-DIAS
     ADD 5                 TO SUM-RES-DIAS
     MOVE SPACES           TO FECHA-VALIDA
                              NUEVA-FECHA-YYYYMMDD

     CALL "SumarDiasFecha" USING SUMARDIAS-ENTRADA
                                 SUMARDIAS-SALIDA

*    AL SUMAR 5 DIAS NUEVA FECHA SERIA 20250719
     DISPLAY "FECHA INICIO   : " FECHA-INICIO
     DISPLAY "NUEVA FECHA    : " NUEVA-FECHA-YYYYMMDD

     MOVE ZEROES           TO SUM-RES-DIAS
     SUBTRACT 20         FROM SUM-RES-DIAS
     MOVE SPACES           TO FECHA-VALIDA
                              NUEVA-FECHA-YYYYMMDD
     CALL "SumarDiasFecha" USING SUMARDIAS-ENTRADA
                                 SUMARDIAS-SALIDA
*    AL RESTAR 20 DIAS NUEVA FECHA SERIA 20250624
     DISPLAY "FECHA INICIO   : " FECHA-INICIO
     DISPLAY "NUEVA FECHA    : " NUEVA-FECHA-YYYYMMDD



      STOP RUN.
 END PROGRAM ProbarSumarDiasFecha.
```

NO COPIAR

ProbarSumarDiasFecha.cbl

```
25          MOVE FECHA-INICIO     TO FECHA-YYYYMMDD
26
27          MOVE ZEROES           TO SUM-RES-DIAS
28          ADD 5                 TO SUM-RES-DIAS
29          MOVE SPACES           TO FECHA-VALIDA
30                                   NUEVA-FECHA-YYYYMMDD
31
32          CALL "SumarDiasFecha" USING SUMARDIAS-ENTRADA
33                                      SUMARDIAS-SALIDA
34
35      *   AL SUMAR 5 DIAS NUEVA FECHA SERIA 20250719
36          DISPLAY "FECHA INICIO   : " FECHA-INICIO
37          DISPLAY "NUEVA FECHA    : " NUEVA-FECHA-YYYYMMDD
38
39          MOVE ZEROES           TO SUM-RES-DIAS
40          SUBTRACT 20         FROM SUM-RES-DIAS
41          MOVE SPACES           TO FECHA-VALIDA
42                                   NUEVA-FECHA-YYYYMMDD
43          CALL "SumarDiasFecha" USING SUMARDIAS-ENTRADA
44                                      SUMARDIAS-SALIDA
45      *   AL RESTAR 20 DIAS NUEVA FECHA SERIA 20250624
46          DISPLAY "FECHA INICIO   : " FECHA-INICIO
47          DISPLAY "NUEVA FECHA    : " NUEVA-FECHA-YYYYMMDD
48
49
```

Logs

Compiler | Issues | Output

```
C:\PgmCobols\Pgms\bin\ProbarSumarDiasFecha.exe
FECHA INICIO   : 20250714
NUEVA FECHA    : 20250719
FECHA INICIO   : 20250714
NUEVA FECHA    : 20250624

Process finished with exit code 0
```

NO C

Diferencia Dias entre 2 Fechas (DiffDiasFecha.CBL)

Para este caso se consideran que entran 2 fechas previamente validada y correctas, la diferencia se obtiene entre la Fecha Inicial y una Fecha Final.

```
******************************************************************
* Author: JORGE DUARTE
* Date: 18-07-2025
* Purpose: ENSEÑANZA
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. DiffDiasFecha.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 ENTERO               PIC 9(04).
     05 ENTERO2              PIC 9(04).
     05 ENTERO3              PIC 9(04).
     05 RESTO                PIC 9(04).
     05 RESTO2               PIC 9(04).
     05 RESTO3               PIC 9(04).
     05 WS-FECHA1.
         10 ANOMES1.
            15 ANO1         PIC 9(04).
            15 MES1         PIC 9(02).
         10 DIA1         PIC 9(02).
     05 WS-FECHA2.
         10 ANOMES2.
            15 ANO2         PIC 9(04).
            15 MES2         PIC 9(02).
         10 DIA2         PIC 9(02).
     05 WS-DIFF           PIC 9(07).
     05 DIAS-X-MES PIC X(24) VALUE "312831303130313130313031".
     05 R-MES REDEFINES DIAS-X-MES OCCURS 12 TIMES.
         10 DIAS-MES     PIC 9(02).

 LINKAGE SECTION.
 01 INP-DIFFDIASFECHA.
     05 FECHA1-YYYYMMDD PIC 9(08).
     05 FECHA2-YYYYMMDD PIC 9(08).
 01 OUT-DIFFDIASFECHA.
     05 DIFF-DIAS        PIC 9(07).

 PROCEDURE DIVISION USING INP-DIFFDIASFECHA
                          OUT-DIFFDIASFECHA.
 MAIN-PROCEDURE.

      MOVE FECHA1-YYYYMMDD   TO WS-FECHA1
      MOVE FECHA2-YYYYMMDD   TO WS-FECHA2

      IF FECHA2-YYYYMMDD < FECHA1-YYYYMMDD THEN
          MOVE 0             TO WS-DIFF
      ELSE
          IF FECHA1-YYYYMMDD = FECHA2-YYYYMMDD THEN
             MOVE 0          TO WS-DIFF
          ELSE
             PERFORM CALCULAR-DIAS
          END-IF
      END-IF
```

NO COPIAR

```
      MOVE WS-DIFF  TO DIFF-DIAS

      GOBACK.

 CALCULAR-DIAS.
     MOVE 0          TO WS-DIFF
     IF ANO1 = ANO2 AND
        MES1 = MES2
        COMPUTE WS-DIFF = DIA2 - DIA1
     ELSE

*    PARTIMOS SUMANDO LA DIFERENCIA ENTRE LA FECHA INICIAL
     COMPUTE WS-DIFF = WS-DIFF + (DIAS-MES(MES1) - DIA1)
*    AGREGAMOS LOS DIAS DE LA FECHA FINAL
     COMPUTE WS-DIFF = WS-DIFF + DIA2

       ADD 1        TO MES1
       IF MES1 > 12 THEN
           MOVE 1  TO MES1
           ADD  1  TO ANO1
       END-IF

       SUBTRACT 1 FROM MES2
       IF MES2 < 1 THEN
          MOVE 12    TO MES2
          SUBTRACT 1 FROM ANO2
       END-IF

       PERFORM UNTIL ANOMES1 > ANOMES2

         IF MES1 = 2 THEN
            DIVIDE ANO1  BY 4   GIVING ENTERO   REMAINDER RESTO
            DIVIDE ANO1  BY 100 GIVING ENTERO2  REMAINDER RESTO2
            DIVIDE ANO1  BY 400 GIVING ENTERO3  REMAINDER RESTO3
            IF RESTO3 = 0 OR (RESTO = 0 AND RESTO2 NOT = 0) THEN
               ADD 29      TO WS-DIFF
            ELSE
               ADD 28      TO WS-DIFF
         ELSE
            ADD DIAS-MES(MES1)   TO WS-DIFF
         END-IF

         ADD 1    TO MES1
         IF MES1 > 12 THEN
             MOVE 1  TO MES1
             ADD  1  TO ANO1
         END-IF

       END-PERFORM
     END-IF
     .

 END PROGRAM DiffDiasFecha.
```

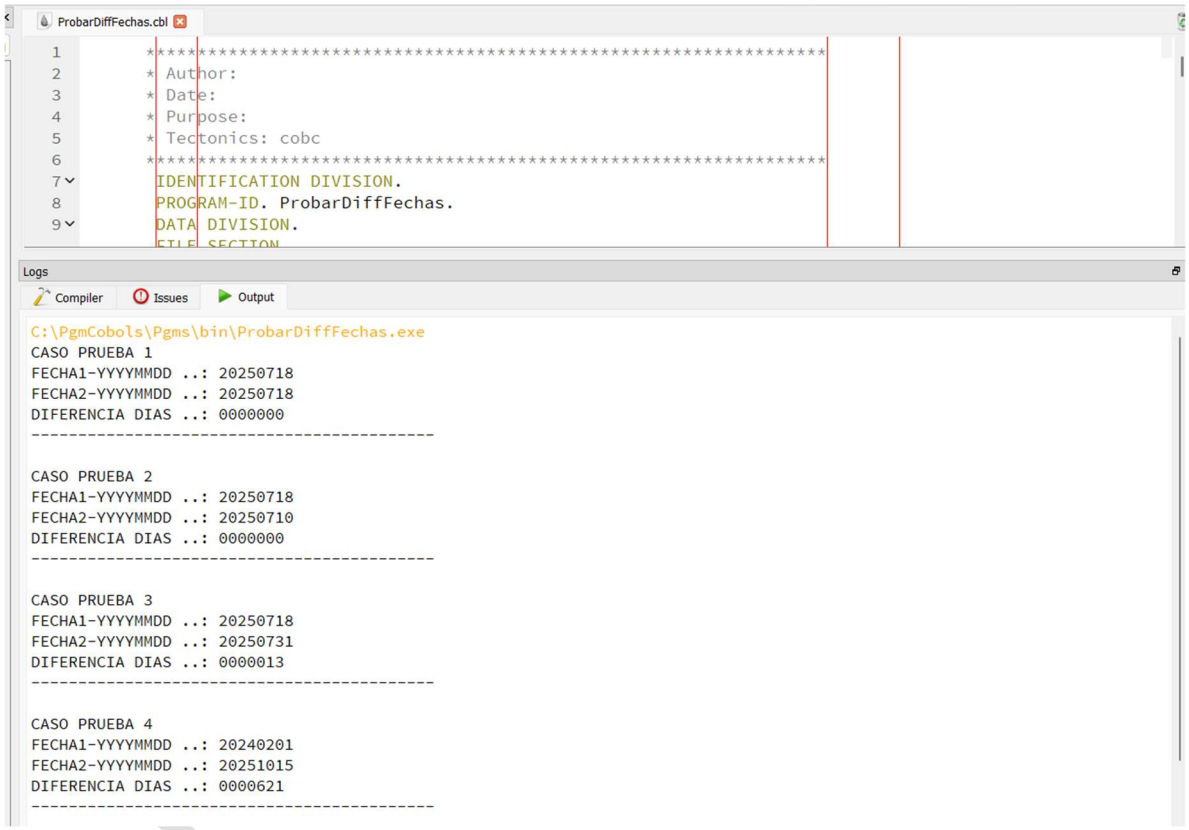
NO COPIAR

Probar Diferencia Dias entre 2 Fechas (ProbarDiffFechas.CBL)

```
******************************************************************
* Author: JORGE DUARTE URZUA
* Date: 18-07-2025
* Purpose: ENSEÑANZA
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ProbarDiffFechas.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 INP-DIFFDIASFECHA.
     05 FECHA1-YYYYMMDD PIC 9(08).
     05 FECHA2-YYYYMMDD PIC 9(08).
 01 OUT-DIFFDIASFECHA.
     05 DIFF-DIAS       PIC 9(07).

 PROCEDURE DIVISION.
 MAIN-PROCEDURE.

* 1- Caso 1 Mostramos la Diferencia entre 2 fechas iguales
     MOVE 20250718     TO FECHA1-YYYYMMDD
     MOVE 20250718     TO FECHA2-YYYYMMDD
     MOVE ZEROES       TO DIFF-DIAS
     CALL "DiffDiasFecha" USING INP-DIFFDIASFECHA
                                OUT-DIFFDIASFECHA

     DISPLAY "CASO PRUEBA 1"
     DISPLAY "FECHA1-YYYYMMDD ..: " FECHA1-YYYYMMDD
     DISPLAY "FECHA2-YYYYMMDD ..: " FECHA2-YYYYMMDD
     DISPLAY "DIFERENCIA DIAS ..: " DIFF-DIAS
     DISPLAY "-----------------------------------------"
     DISPLAY " "

* 2- Caso 2 Fecha Hasta < Fecha Desde
     MOVE 20250718     TO FECHA1-YYYYMMDD
     MOVE 20250710     TO FECHA2-YYYYMMDD
     MOVE ZEROES       TO DIFF-DIAS
     CALL "DiffDiasFecha" USING INP-DIFFDIASFECHA
                                OUT-DIFFDIASFECHA

     DISPLAY "CASO PRUEBA 2"
     DISPLAY "FECHA1-YYYYMMDD ..: " FECHA1-YYYYMMDD
     DISPLAY "FECHA2-YYYYMMDD ..: " FECHA2-YYYYMMDD
     DISPLAY "DIFERENCIA DIAS ..: " DIFF-DIAS
     DISPLAY "-----------------------------------------"
     DISPLAY " "

* 3- Caso 3 Fecha en el mismo mes
     MOVE 20250718     TO FECHA1-YYYYMMDD
     MOVE 20250731     TO FECHA2-YYYYMMDD
     MOVE ZEROES       TO DIFF-DIAS
     CALL "DiffDiasFecha" USING INP-DIFFDIASFECHA
                                OUT-DIFFDIASFECHA

     DISPLAY "CASO PRUEBA 3"
     DISPLAY "FECHA1-YYYYMMDD ..: " FECHA1-YYYYMMDD
     DISPLAY "FECHA2-YYYYMMDD ..: " FECHA2-YYYYMMDD
     DISPLAY "DIFERENCIA DIAS ..: " DIFF-DIAS
     DISPLAY "-----------------------------------------"
     DISPLAY " "
```

NO COPIAR

```
* 4- Caso 4
     MOVE 20240201     TO FECHA1-YYYYMMDD
     MOVE 20251015     TO FECHA2-YYYYMMDD
     MOVE ZEROES       TO DIFF-DIAS
     CALL "DiffDiasFecha" USING INP-DIFFDIASFECHA
                                OUT-DIFFDIASFECHA

     DISPLAY "CASO PRUEBA 4"
     DISPLAY "FECHA1-YYYYMMDD ..: " FECHA1-YYYYMMDD
     DISPLAY "FECHA2-YYYYMMDD ..: " FECHA2-YYYYMMDD
     DISPLAY "DIFERENCIA DIAS ..: " DIFF-DIAS
     DISPLAY "-------------------------------------------"
     DISPLAY " "

      STOP RUN.
 END PROGRAM ProbarDiffFechas.
```



Calcular Largo String Máx 10000 Caracteres (LargoString10000.CBL)

```
******************************************************************
* Author: JORGE DUARTE
* Date: 08-07-2025
* Purpose: Eseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. LargoString10000.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 LARGO-STR PIC 9(05).

 LINKAGE SECTION.
 01 INPUT-LARGOSTR10000.
     05 STRING-CALCULAR   PIC X(10000).
 01 OUTPUT-LARGOSTR10000.
     05 LK-LARGO-STR     PIC 9(05).

 PROCEDURE DIVISION USING INPUT-LARGOSTR10000
                          OUTPUT-LARGOSTR10000.

 MAIN-PROCEDURE.

     PERFORM VARYING LARGO-STR FROM 10000 BY -1 UNTIL
             LARGO-STR = 0 OR
             STRING-CALCULAR(LARGO-STR:1) NOT = SPACES
     END-PERFORM

     MOVE LARGO-STR   TO LK-LARGO-STR

     GOBACK
     .

 END PROGRAM LargoString10000.
```

NO COPIAR

Validar Tarjeta de Crédito (ValidaTarjetaCredito.CBL)

```
******************************************************************
    * Author: JORGE DUARTE
    * Date: 14-07-2025
    * Purpose: Enseñanza - Utiliza Algoritmo de Luhn
    * Tectonics: cobc
    ******************************************************************
     IDENTIFICATION DIVISION.
     PROGRAM-ID. ValidaTarjetaCredito.
     DATA DIVISION.
     FILE SECTION.
     WORKING-STORAGE SECTION.
     01 VARIABLES-DE-TRABAJO.
         05 WS-SUMA    PIC S9(7).
         05 WS-MULT    PIC S9(2).
         05 I          PIC 9(02).
         05 ENTERO     PIC 9(3).
         05 RESTO      PIC 9(3).
         05 WS-DIGITO  PIC 9.


     LINKAGE SECTION.
     01 PAR-INPUT.
         05 LK-TARJETA    PIC X(16).
     01 PAR-OUTPUT.
         05 LK-CODRET      PIC X.
    *       S = Tarjeta Válida
    *       N = Tarjeta Inválida
     PROCEDURE DIVISION USING PAR-INPUT
                              PAR-OUTPUT.
     MAIN-PROCEDURE.

         MOVE ZEROES         TO WS-SUMA
         MOVE "N"            TO LK-CODRET
         MOVE "4830310043224451" TO LK-TARJETA
         PERFORM VARYING I FROM 1 BY 1 UNTIL I > 16
            IF LK-TARJETA(I:1) IS NUMERIC THEN
                MOVE LK-TARJETA(I:1) TO WS-DIGITO
                DIVIDE I BY 2 GIVING ENTERO
                                  REMAINDER RESTO
                IF RESTO = 0 THEN
                   ADD WS-DIGITO  TO WS-SUMA
                ELSE
                   COMPUTE WS-MULT = WS-DIGITO * 2
                   IF WS-MULT > 9 THEN
                       SUBTRACT 9 FROM WS-MULT
                   END-IF
                   ADD WS-MULT   TO WS-SUMA
                END-IF
            ELSE
    *           SI EL DIGITO NO ES UN NUMERO DEVOLVEMOS INMEIDATAMENTE
    *           QUE LA TARJETA NO ES VALIDA
                ADD 26 TO I
            END-IF
         END-PERFORM

         IF I <= 17 THEN
             DIVIDE WS-SUMA BY 10 GIVING ENTERO
                                  REMAINDER RESTO
             IF RESTO = 0 THEN
                MOVE "S"    TO LK-CODRET
             END-IF
```

NO COPIAR

```
    END-IF

    GOBACK.
END PROGRAM ValidaTarjetaCredito.
```

NO COPIAR

Calcular Edad (CalcularEdad.CBL)

```
      ******************************************************************
      * Author: JORGE DUARTE
      * Date: 16-06-2025
      * Purpose: ENSEÑANZA
      * Tectonics: cobc
      ******************************************************************
       IDENTIFICATION DIVISION.
       PROGRAM-ID. CalcularEdad.
       DATA DIVISION.
       FILE SECTION.
       WORKING-STORAGE SECTION.

       01 VARIABLES-DE-TRABAJO.
           05 ENTERO               PIC 9(04).
           05 ENTERO2              PIC 9(04).
           05 ENTERO3              PIC 9(04).
           05 RESTO                PIC 9(04).
           05 RESTO2               PIC 9(04).
           05 RESTO3               PIC 9(04).
           05  FECHA1              PIC 9(8).
           05  R-FECHA1 REDEFINES FECHA1.
               10  ANO1                PIC 9(4).
               10  MES1                PIC 9(2).
               10  DIA1                PIC 9(2).
           05  FECHA2              PIC 9(8).
           05 R-FECHA2 REDEFINES FECHA2.
               10  ANO2                PIC 9(4).
               10  MES2                PIC 9(2).
               10  DIA2                PIC 9(2).
           05 WS-EDAD-ANOS     PIC S9(04).
           05 WS-EDAD-MESES    PIC S9(04).
           05 WS-EDAD-DIAS     PIC S9(02).
           05 CANTIDAD-DIA-MES PIC 9(02).


       LINKAGE SECTION.
       01 INP-CALCULA-EDAD.
           05 FECHA1-YYYYMMDD PIC X(08).
      *      |
      *      +--> Fecha Inicio a Calcular por Ejemplo Fecha de Nacimiento
           05 FECHA2-YYYYMMDD PIC X(08).
      *      |
      *      +--> Fecha contra la cual calcular, generalmente la fecha Actual

       01 OUT-CALCULA-EDAD.
           05 OUT-CODRET-EDAD PIC 9(02).
      *       0 = Fecha2 es Mayor o Igual a Fecha1
      *       1 = No se puede calcular Edad
           05 OUT-EDAD-ANOS   PIC 9(04).
           05 OUT-EDAD-MESES  PIC 9(04).
           05 OUT-EDAD-DIAS   PIC 9(02).

       PROCEDURE DIVISION USING INP-CALCULA-EDAD
                                OUT-CALCULA-EDAD.

       MAIN-PROCEDURE.

           MOVE 0                TO OUT-CODRET-EDAD
           MOVE FECHA1-YYYYMMDD  TO FECHA1
           MOVE FECHA2-YYYYMMDD  TO FECHA2
           IF FECHA2 >= FECHA1 THEN
```

NO COPIAR

```
           PERFORM CALCULAR-EDAD
       ELSE
           MOVE 1   TO OUT-CODRET-EDAD
       END-IF

       GOBACK.

 CALCULAR-EDAD.
*--------------
       MOVE    ZEROES           TO WS-EDAD-ANOS
                                   WS-EDAD-MESES
                                   WS-EDAD-DIAS

       COMPUTE WS-EDAD-ANOS  = ANO2 - ANO1
       COMPUTE WS-EDAD-MESES = MES2 - MES1
       COMPUTE WS-EDAD-DIAS  = DIA2 -  DIA1

       IF WS-EDAD-DIAS < 0 THEN
          SUBTRACT 1 FROM WS-EDAD-MESES
          PERFORM DIAS-MES
          ADD CANTIDAD-DIA-MES TO WS-EDAD-DIAS
       END-IF

       IF WS-EDAD-MESES < 0 THEN
          SUBTRACT 1    FROM WS-EDAD-ANOS
          ADD 12          TO WS-EDAD-MESES
       END-IF

       MOVE WS-EDAD-ANOS   TO OUT-EDAD-ANOS
       MOVE WS-EDAD-MESES  TO OUT-EDAD-MESES
       MOVE WS-EDAD-DIAS   TO OUT-EDAD-DIAS

       .
 DIAS-MES.

       MOVE   31   TO CANTIDAD-DIA-MES
       IF MES1 = 2 THEN
           DIVIDE   ANO1  BY 4   GIVING ENTERO   REMAINDER RESTO
           DIVIDE   ANO1  BY 100 GIVING ENTERO2  REMAINDER RESTO2
           DIVIDE   ANO1  BY 400 GIVING ENTERO3  REMAINDER RESTO3
           IF RESTO3 = 0 OR (RESTO = 0 AND RESTO2 NOT = 0) THEN
               MOVE 29      TO CANTIDAD-DIA-MES
           ELSE
               MOVE 28      TO CANTIDAD-DIA-MES
           END-IF
       ELSE
           IF MES1 = 4 OR 6 OR 9 OR 11 THEN
               MOVE 30      TO CANTIDAD-DIA-MES
           END-IF
       END-IF
       .
 END PROGRAM CalcularEdad.
```

NO COPIAR

Buscar Posición de un Carácter (PosCarStr.CBL)

```
******************************************************************
* Author: JORGE DUARTE URZUA.
* Date: 16-07-2025
* Purpose: ENSEÑANZA
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. PosCarString.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 77 I                    PIC 9(05).
 77 SALIR                PIC X.

 01 INPUT-LARGOSTR10000.
     05 STRING-CALCULAR   PIC X(10000).
 01 OUTPUT-LARGOSTR10000.
     05 LK-LARGO-STR     PIC 9(05).

 LINKAGE SECTION.
 01 INP-POSCARSTR.
     05 POSCARSTR-STRING-BUSCAR    PIC X(10000).
     05 POSCARSTR-CARACTER-BUSCAR  PIC X.
 01 OUT-POSCARSTR.
     05 POSCARSTR-POSICION         PIC 9(05).

 PROCEDURE DIVISION USING INP-POSCARSTR
                          OUT-POSCARSTR.
 MAIN-PROCEDURE.

     MOVE POSCARSTR-STRING-BUSCAR TO STRING-CALCULAR
     MOVE ZEROES                  TO LK-LARGO-STR
     CALL "LargoString10000" USING INPUT-LARGOSTR10000
                                   OUTPUT-LARGOSTR10000
     MOVE 0   TO I
     MOVE "N" TO SALIR
     PERFORM  UNTIL I > LK-LARGO-STR OR SALIR = 'S'
         ADD 1    TO I
         IF POSCARSTR-STRING-BUSCAR(I:1) =
            POSCARSTR-CARACTER-BUSCAR THEN
            MOVE "S"    TO SALIR
         END-IF

     END-PERFORM

     IF SALIR = "S" THEN
        MOVE I    TO POSCARSTR-POSICION
     ELSE
        MOVE 0    TO POSCARSTR-POSICION
     END-IF


     GOBACK.
 END PROGRAM PosCarString.
```

NO COPIAR

Centrar Texto (CentrarTexto.CBL)

```
******************************************************************
* Author: JORGE DUARTE
* Date: 16-07-2025
* Purpose: ENSEÑANZA
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. CentrarTexto.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 POS-STR          PIC 9(3).

 01 PAR-INPUT-LARGOSTR.
     05 STRING-CALCULAR   PIC X(500).
 01 PAR-OUTPUT-LARGOSTR.
     05 LK-LARGO-STR     PIC 9(03).

 LINKAGE SECTION.
 01 INP-CENTRAR-TXT.
     05 TEXTO-CENTRAR PIC X(100).
 01 OUT-CENTRAR-TXT.
     05 TXT-CODRET     PIC X.
*       S = TEXTO TEXTO-CENTRADO
*       N = NO SE PUDO CENTRAR TEXTO
     05 TEXTO-CENTRADO PIC X(100).

 PROCEDURE DIVISION USING INP-CENTRAR-TXT
                          OUT-CENTRAR-TXT.
 MAIN-PROCEDURE.

      MOVE TEXTO-CENTRAR   TO STRING-CALCULAR
      MOVE 0               TO LK-LARGO-STR
      CALL "LargoString"  USING PAR-INPUT-LARGOSTR
                                PAR-OUTPUT-LARGOSTR

      IF LK-LARGO-STR < 1 THEN
         MOVE "N"           TO TXT-CODRET
         MOVE TEXTO-CENTRAR TO TEXTO-CENTRADO
      ELSE
         COMPUTE POS-STR = (100 - LK-LARGO-STR ) / 2
         MOVE SPACES         TO TEXTO-CENTRADO
         MOVE TEXTO-CENTRAR(1:LK-LARGO-STR)  TO
                               TEXTO-CENTRADO(POS-STR:)
         MOVE "S"            TO TXT-CODRET
      END-IF

      GOBACK.
 END PROGRAM CentrarTexto.
```

NO COPIAR

Validar Email (ValidarEmail.CBL)

```
******************************************************************
* Author: JORGE DUARTE
* Date: 17-07-2025
* Purpose: ENSEÑANZA
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
     PROGRAM-ID. ValidarEmail.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.

 01 VARIABLES-DE-TRABAJO.
     05 CUENTA-1     PIC 9(03).
     05 CUENTA-2     PIC 9(03).
     05 WS-ANTES-DE-ARROBA   PIC X(150).
     05 WS-DESPUES-DE-ARROBA PIC X(150).

 LINKAGE SECTION.
 01 INP-VALIDA-EMAIL.
     05 WS-EMAIL     PIC X(150).
 01 OUT-VALIDA-EMAIL.
     05 MSG-ERROR   PIC X(50).

 PROCEDURE DIVISION USING INP-VALIDA-EMAIL
                          OUT-VALIDA-EMAIL.
 MAIN-PROCEDURE.
     MOVE "EMAIL CORRECTO "     TO MSG-ERROR
     MOVE ZEROES                TO CUENTA-1
                                   CUENTA-2


     IF WS-EMAIL = SPACES
        MOVE "DEBE INGRESAR EMAIL " TO MSG-ERROR
     ELSE
        INSPECT WS-EMAIL TALLYING CUENTA-1 FOR ALL "@"

        IF CUENTA-1 = 1 THEN
           UNSTRING WS-EMAIL DELIMITED BY "@"
           INTO     WS-ANTES-DE-ARROBA,
                    WS-DESPUES-DE-ARROBA

           IF WS-ANTES-DE-ARROBA(1:1) NOT = SPACES AND
              WS-ANTES-DE-ARROBA(2:1) NOT = SPACES AND
              WS-ANTES-DE-ARROBA(3:1) NOT = SPACES
               THEN
              INSPECT WS-DESPUES-DE-ARROBA TALLYING
              CUENTA-2 FOR ALL "."
              IF CUENTA-2 = 0 THEN
                 MOVE
                 "DEBE INGRESAR EMAIL CON FORMATO CORRECTO "
                                    TO MSG-ERROR
              ELSE
                IF WS-DESPUES-DE-ARROBA(1:1) = SPACES AND
                   WS-DESPUES-DE-ARROBA(2:1) = SPACES AND
                   WS-DESPUES-DE-ARROBA(3:1) = SPACES
                    THEN
                   MOVE
                    "DEBE INGRESAR EMAIL CON FORMATO CORRECTO "
                   TO MSG-ERROR
                END-IF
              END-IF
```

NO COPIAR

```
            ELSE
               MOVE "DEBE INGRESAR EMAIL CON FORMATO CORRECTO"
                                        TO MSG-ERROR
            END-IF

         ELSE
            MOVE "DEBE INGRESAR EMAIL CON FORMATO CORRECTO"
                                     TO MSG-ERROR
         END-IF

      END-IF
      GOBACK
      .

END PROGRAM ValidarEmail.
```

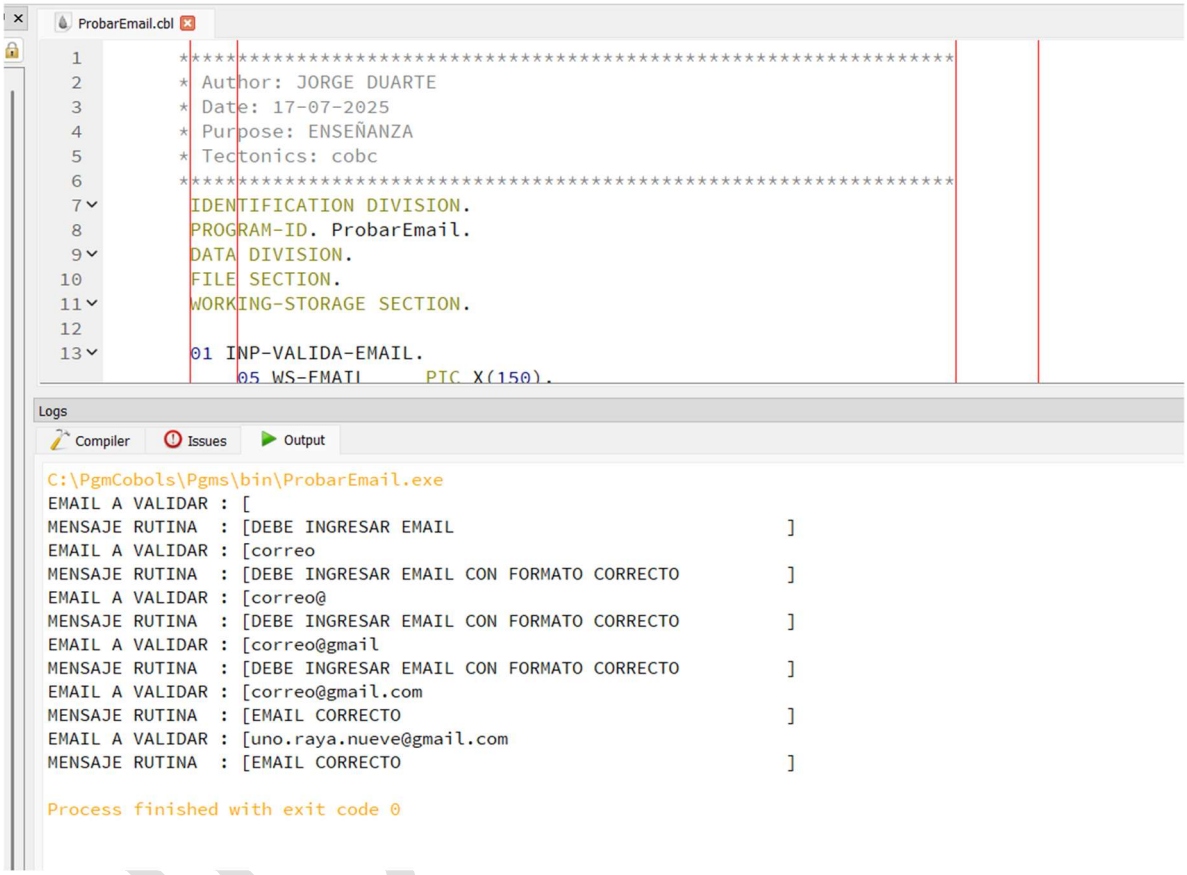
NO COPIAR

Probar Validar Email (ProbarEmail.CBL)

```
******************************************************************
* Author: JORGE DUARTE
* Date: 17-07-2025
* Purpose: ENSEÑANZA
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ProbarEmail.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.

 01 INP-VALIDA-EMAIL.
     05 WS-EMAIL     PIC X(150).
 01 OUT-VALIDA-EMAIL.
     05 MSG-ERROR   PIC X(50).

 PROCEDURE DIVISION.
 MAIN-PROCEDURE.
      MOVE SPACES           TO WS-EMAIL
      MOVE SPACES           TO MSG-ERROR
      CALL "ValidarEmail" USING INP-VALIDA-EMAIL
                                OUT-VALIDA-EMAIL

      DISPLAY "EMAIL A VALIDAR : [" WS-EMAIL "]"
      DISPLAY "MENSAJE RUTINA  : [" MSG-ERROR "]"

      MOVE "correo"          TO WS-EMAIL
      MOVE SPACES            TO MSG-ERROR
      CALL "ValidarEmail" USING INP-VALIDA-EMAIL
                                OUT-VALIDA-EMAIL

      DISPLAY "EMAIL A VALIDAR : [" WS-EMAIL "]"
      DISPLAY "MENSAJE RUTINA  : [" MSG-ERROR "]"

      MOVE "correo@"         TO WS-EMAIL
      MOVE SPACES            TO MSG-ERROR
      CALL "ValidarEmail" USING INP-VALIDA-EMAIL
                                OUT-VALIDA-EMAIL

      DISPLAY "EMAIL A VALIDAR : [" WS-EMAIL "]"
      DISPLAY "MENSAJE RUTINA  : [" MSG-ERROR "]"

      MOVE "correo@gmail"      TO WS-EMAIL
      MOVE SPACES              TO MSG-ERROR
      CALL "ValidarEmail" USING INP-VALIDA-EMAIL
                                OUT-VALIDA-EMAIL

      DISPLAY "EMAIL A VALIDAR : [" WS-EMAIL "]"
      DISPLAY "MENSAJE RUTINA  : [" MSG-ERROR "]"

      MOVE "correo@gmail.com"      TO WS-EMAIL
      MOVE SPACES                  TO MSG-ERROR
      CALL "ValidarEmail" USING INP-VALIDA-EMAIL
                                OUT-VALIDA-EMAIL

      DISPLAY "EMAIL A VALIDAR : [" WS-EMAIL "]"
      DISPLAY "MENSAJE RUTINA  : [" MSG-ERROR "]"

      MOVE "uno.raya.nueve@gmail.com"      TO WS-EMAIL
      MOVE SPACES                          TO MSG-ERROR
```

NO COPIAR

```
           CALL "ValidarEmail" USING INP-VALIDA-EMAIL
                                     OUT-VALIDA-EMAIL

           DISPLAY "EMAIL A VALIDAR : [" WS-EMAIL "]"
           DISPLAY "MENSAJE RUTINA  : [" MSG-ERROR "]"


           STOP RUN.
       END PROGRAM ProbarEmail.
```


ProbarEmail.cbl
1 **
2 * Author: JORGE DUARTE
3 * Date: 17-07-2025
4 * Purpose: ENSEÑANZA
5 * Tectonics: cobc
6 **
7 IDENTIFICATION DIVISION.
8 PROGRAM-ID. ProbarEmail.
9 DATA DIVISION.
10 FILE SECTION.
11 WORKING-STORAGE SECTION.
12
13 01 INP-VALIDA-EMAIL.
05 WS-EMAIL PIC X(150).
Logs
Compiler
Issues
Output
C:\PgmCobols\Pgms\bin\ProbarEmail.exe
EMAIL A VALIDAR : [
MENSAJE RUTINA : [DEBE INGRESAR EMAIL]
EMAIL A VALIDAR : [correo
MENSAJE RUTINA : [DEBE INGRESAR EMAIL CON FORMATO CORRECTO]
EMAIL A VALIDAR : [correo@
MENSAJE RUTINA : [DEBE INGRESAR EMAIL CON FORMATO CORRECTO]
EMAIL A VALIDAR : [correo@gmail
MENSAJE RUTINA : [DEBE INGRESAR EMAIL CON FORMATO CORRECTO]
EMAIL A VALIDAR : [correo@gmail.com
MENSAJE RUTINA : [EMAIL CORRECTO]
EMAIL A VALIDAR : [uno.raya.nueve@gmail.com
MENSAJE RUTINA : [EMAIL CORRECTO]
Process finished with exit code 0

Leer un Archivo (LeerArchivo.CBL)

Para poder ejecutar este programa necesitamos un archivo por ejemplo “cargar.txt”, el cual para nuestro ejemplo, cada campo debe estar separado por “;” y debe poseer la siguiente estructura como ejemplo (se puede alterar de acuerdo a las necesidades).

```
1;9:UNO;RAYA;NUEVE;+5612345678;correo@gmail.com;H;01-01-
1973;Chileno;Alameda;Santiago

2;7:DOS;RAYA;SIETE;+5698765432;dos_raya_siete@hotmail.com;M;25-08-
1940;Argentino;Providencia;Santiago
```

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. LeerArchivo.
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT archivo-entrada ASSIGN TO
           "C:\PgmCobols\Data\cargar.txt"
               ORGANIZATION IS LINE SEQUENTIAL.

       DATA DIVISION.
       FILE SECTION.
       FD  archivo-entrada.
       01  registro-entrada      PIC X(255).

       WORKING-STORAGE SECTION.
       01  RUT                   PIC 9(10).
       01  DV                    PIC X(1).
       01  NOMBRE                PIC X(30).
       01  APEPAT                PIC X(30).
       01  APEMAT                PIC X(30).
       01  TELEFONO              PIC X(15).
       01  EMAIL                 PIC X(100).
       01  GENERO                PIC X(01).
       01  FECNAC                PIC X(10).
       01  NACIONALIDAD          PIC X(30).
       01  DIRECCION             PIC X(100).
       01  CIUDAD                PIC X(50).

       01  fin-lectura           PIC X VALUE 'N'.

       PROCEDURE DIVISION.
       INICIO.
           OPEN INPUT archivo-entrada
           PERFORM UNTIL fin-lectura = 'S'
               READ archivo-entrada INTO registro-entrada
                   AT END
                       MOVE 'S' TO fin-lectura
                   NOT AT END
                       PERFORM separar-campos
               END-READ
           END-PERFORM
           CLOSE archivo-entrada
           DISPLAY "Fin de la lectura."
           STOP RUN.
```

NO COPIAR

```
SEPARAR-CAMPOS.
    MOVE registro-entrada TO RUT
    UNSTRING registro-entrada DELIMITED BY ";"
        INTO RUT, DV, NOMBRE, APEPAT, APEMAT,
             TELEFONO, EMAIL, GENERO, FECNAC,
             NACIONALIDAD, DIRECCION, CIUDAD

    END-UNSTRING
    DISPLAY "RUT: " RUT
    DISPLAY "DV: " DV
    DISPLAY "Nombre: " NOMBRE
    DISPLAY "Apellido Paterno: " APEPAT
    DISPLAY "Apellido Materno: " APEMAT
    DISPLAY "Telefono: " TELEFONO
    DISPLAY "Email: " EMAIL
    DISPLAY "Genero: " GENERO
    DISPLAY "Fecha de Nacimiento: " FECNAC
    DISPLAY "Nacionalidad: " NACIONALIDAD
    DISPLAY "Direccion: " DIRECCION
    DISPLAY "Ciudad: " CIUDAD
    DISPLAY "-------------------------".
```

LeerArchivo.cbl

```
1   IDENTIFICATION DIVISION.
2   PROGRAM-ID. LeerArchivo.
3   ENVIRONMENT DIVISION.
4   INPUT-OUTPUT SECTION.
5   FILE-CONTROL.
6       SELECT archivo-entrada ASSIGN TO
7       "C:\PgmCobols\Data\cargar.txt"
8           ORGANIZATION IS LINE SEQUENTIAL.
9
10  DATA DIVISION.
11  FILE SECTION.
12  FD  archivo-entrada.
13  01  registro-entrada      PIC X(255).
14
15  WORKING-STORAGE SECTION.
16  01  RUT                   PIC 9(10).
    01  DV                    PIC X(1)
```

Logs

Compiler | Issues | Output

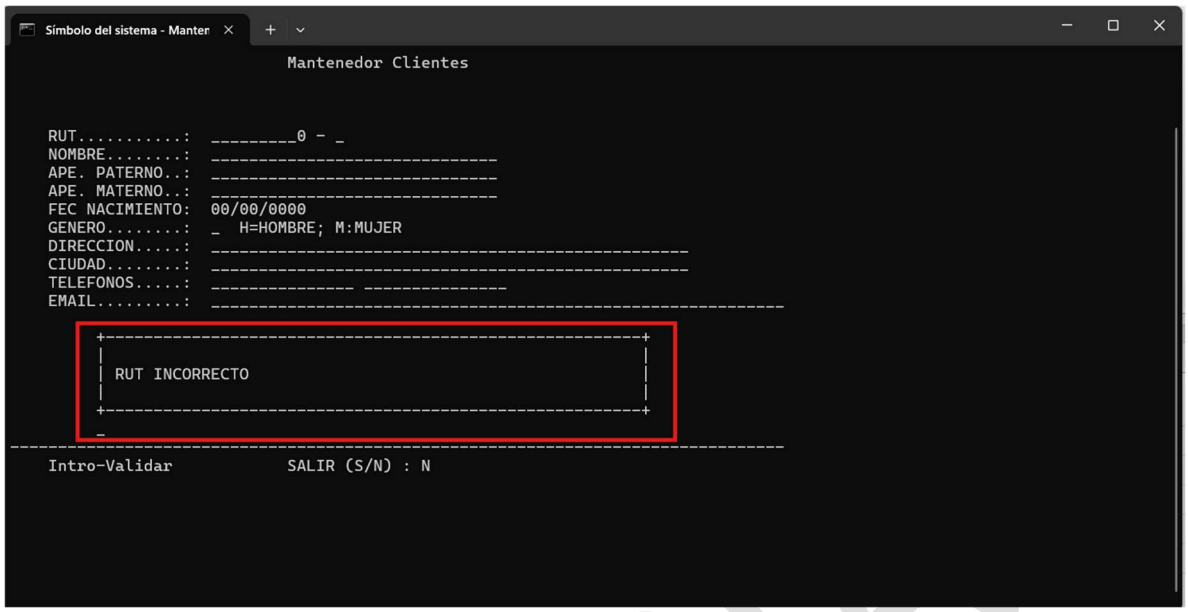
```
C:\PgmCobols\Pgms\bin\LeerArchivo.exe
RUT: 0000000001
DV: 9
Nombre: RAYA
Apellido Paterno: NUEVE
Apellido Materno: +5612345678
Telefono: correo@gmail.co
Email: H
Genero: 0
Fecha de Nacimiento: Chileno
Nacionalidad: Alameda
Direccion: Santiago
Ciudad:
-------------------------
RUT: 0000000002
DV: 7
Nombre: RAYA
```

Mensaje (VentanaMsj.CBL)

El siguiente programa permite mostrar un mensaje de 1 línea, el cual puede ser modificado para mostrar más linea y cambiar donde se muestra el mensaje, este tipo de programas o rutinas es útil cuando se quieren desplegar varios mensajes a la vez.

```
******************************************************************
* Author:JORGE DUARTE
* Date: 07-07-2025
* Purpose: Enseñanza
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. VentanaMsj.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.

 01 WS-ENTER PIC X.



 LINKAGE SECTION.
 01 MENSAJE-MOSTRAR PIC X(50).

 SCREEN SECTION.
 01  PANTALLA-VENTANA.
    03 LINE 16 COLUMN 10 VALUE
     "+------------------------------------------------------+".
    03 LINE 17 COLUMN 10 VALUE
     "|                                                      |".
    03 LINE 18 COLUMN 10 VALUE
     "| ".
    03 LINE 18 COLUMN 12 PIC X(50) FROM MENSAJE-MOSTRAR.
    03 LINE 18 COLUMN 63 VALUE
     "    |".
    03 LINE 19 COLUMN 10 VALUE
     "|                                                      |".
    03 LINE 20 COLUMN 10 VALUE
     "+------------------------------------------------------+".
    03 LINE 21 COLUMN 10 PIC X USING WS-ENTER.

 PROCEDURE DIVISION USING MENSAJE-MOSTRAR.
 MAIN-PROCEDURE.

     DISPLAY PANTALLA-VENTANA
     ACCEPT  PANTALLA-VENTANA

    GOBACK.

 END PROGRAM VentanaMsj.
```

NO COPIAR


Símbolo del sistema - Manter
Mantenedor Clientes
RUT.....: _____0 - _
NOMBRE.....: _____
APE. PATERNO..: _____
APE. MATERNO..: _____
FEC NACIMIENTO: 00/00/0000
GENERO.....: _ H=HOMBRE; M:MUJER
DIRECCION.....: _____
CIUDAD.....: _____
TELEFONOS.....: _____ _____
EMAIL.....: _____
RUT INCORRECTO
Intro-Validar
SALIR (S/N) : N

NO COPIAR

Grabar en un Archivo (GrabarCliente.CBL)

Para nuestro ejemplo vamos a simular crear un Cliente, para ello requeriremos que exista previante creado:

- Archivo C:\PgmCobols\Data\clientes.dat
- Tener la estructura C:\PgmCobols\FD\FDClientes.cbl

```
01  usuario-registro.
    05  usuario-rut        PIC 9(10).
    05  usuario-dv         PIC X.
    05  usuario-nombre     PIC X(30).
    05  usuario-apepat     PIC X(30).
    05  usuario-apemat     PIC X(30).
    05  usuario-fecnac     PIC 9(08).
    05  usuario-genero     PIC X.
    05  usuario-direccion  PIC X(50).
    05  usuario-ciudad     PIC X(50).
    05  usuario-telefono1  PIC X(15).
    05  usuario-telefono2  PIC X(15).
    05  usuario-email      PIC X(100).
    05  usuario-feccre     PIC 9(08).
```

- El programa recibe como parámetro todos los campos, lo hicimos esta forma para que se entienda el propósito.
- La ejecución de este programa se realizará cuando desarrollemos el Mantenedor.cbl.

```
******************************************************************
* Author: Jorge Duarte
* Date: 02-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. GrabarCliente.

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT datos-usuarios
         ASSIGN TO
     "C:\PgmCobols\Data\clientes.dat"
         ORGANIZATION IS INDEXED
         ACCESS MODE IS DYNAMIC
         RECORD KEY IS usuario-rut
         FILE STATUS IS FS-USUARIOS.

 DATA DIVISION.
 FILE SECTION.
 FD  datos-usuarios.
     copy "C:\PgmCobols\FD\FDCliente.cbl".

 WORKING-STORAGE SECTION.
 01  fin-de-archivo      PIC X VALUE 'N'.
 01  usuario-clave       PIC X(10).
```

NO COPIAR

```
01  FS-USUARIOS         PIC XX.


01 VARIABLES-PEDIR-PANTALLA.

    05 RUT        PIC 9(10).
    05 DV         PIC X.
    05 NOMBRE     PIC X(30).
         05 APEPAT     PIC X(30).
         05 APEMAT     PIC X(30).
         05 FECNAC     PIC 9(08).
    05 R-FECNAC REDEFINES FECNAC.
       10 DIA-NAC PIC 9(02).
       10 MES-NAC PIC 9(02).
       10 ANO-NAC PIC 9(04).
         05 GENERO     PIC X.
    05 DIRECCION  PIC X(50).
         05 CIUDAD     PIC X(50).
         05 TELEFONO1  PIC X(15).
         05 TELEFONO2  PIC X(15).
         05 EMAIL      PIC X(100).
         05 FECCRE     PIC 9(08).
    05 SALIR      PIC X.
    05 MSG-ERROR  PIC X(70).

01 PARAMETROS-VALIDA-FECHA.
    05 PAR-INP-FECHA.
        10 FECHA-X     PIC X(08).
        10 FORMATO-X   PIC X VALUE "2".
    05 PAR-OUT-FECHA.
        10 FECHA-VALIDA PIC X.

01 PARAMETROS-LARGO-STRING.
    05 STRING-CALCULAR   PIC X(500).
    05 STRING-LARGO      PIC 9(03).

01 VARIABLES-WS-PANTALLA.

    05 WS-RUT        PIC 9(10).
    05 WS-DV         PIC X.
    05 WS-NOMBRE     PIC X(30).
    05 WS-APEPAT     PIC X(30).
    05 WS-APEMAT     PIC X(30).
    05 WS-FECNAC     PIC 9(08).
    05 WS-R-FECNAC REDEFINES WS-FECNAC.
       10 WS-DIA-NAC PIC 9(02).
       10 WS-MES-NAC PIC 9(02).
       10 WS-ANO-NAC PIC 9(04).
    05 WS-GENERO     PIC X.
    05 WS-DIRECCION  PIC X(50).
    05 WS-CIUDAD     PIC X(50).
    05 WS-TELEFONO1  PIC X(15).
    05 WS-TELEFONO2  PIC X(15).
    05 WS-EMAIL      PIC X(100).
    05 WS-FECCRE     PIC 9(08).
    05 WS-SALIR      PIC X.

LINKAGE SECTION.
01 PAR-ENTRADA.
    05  lk-rut        PIC 9(10).
    05  lk-dv         PIC X.
    05  lk-nombre     PIC X(30).
    05  lk-apepat     PIC X(30).
```

NO COPIAR

```
    05  lk-apemat     PIC X(30).
    05  lk-fecnac     PIC 9(08).
    05  lk-genero     PIC X.
    05  lk-direccion  PIC X(50).
    05  lk-ciudad     PIC X(50).
    05  lk-telefono1  PIC X(15).
    05  lk-telefono2  PIC X(15).
    05  lk-email      PIC X(100).
    05  lk-feccre     PIC 9(08).
01 PAR-SALIDA.
    05  OUT-CODRET   PIC 9(04).
    05  OUT-DESRET   PIC X(50).


PROCEDURE DIVISION USING PAR-ENTRADA
                         PAR-SALIDA.

    OPEN I-O datos-usuarios

    MOVE PAR-ENTRADA   TO usuario-registro

    READ datos-usuarios END-READ
    IF FS-USUARIOS = "00"
       MOVE 1    TO OUT-CODRET
       MOVE "USUARIO YA EXISTE" TO OUT-DESRET
    ELSE
        WRITE usuario-registro END-WRITE
        IF FS-USUARIOS = "00"
           MOVE 0                    TO OUT-CODRET
           MOVE "REGISTRO INSERTADO" TO OUT-DESRET
        ELSE
           MOVE 2       TO OUT-CODRET
           MOVE SPACES TO OUT-DESRET
           STRING "ERROR AL INSERTAR (" DELIMITED SIZE
                  FS-USUARIOS           DELIMITED SIZE
                  ")"                   DELIMITED SIZE
                                   INTO OUT-DESRET

        END-IF
    END-IF
    CLOSE datos-usuarios

    GOBACK
    .
END PROGRAM GrabarCliente.
```

NO COPIAR

Borrar un Registro de un Archivo (BorrarCliente.CBL)

Para nuestro caso seguimos el ejercicio anterior, sin embargo en este caso sólo recibiremos como campo de entrada el campo KEY del Archivo.

- Archivo C:\PgmCobols\Data\clientes.dat
- Tener la estructura C:\PgmCobols\FD\FDClientes.cbl

```
01  usuario-registro.
    05  usuario-rut        PIC 9(10).
    05  usuario-dv         PIC X.
    05  usuario-nombre     PIC X(30).
    05  usuario-apepat     PIC X(30).
    05  usuario-apemat     PIC X(30).
    05  usuario-fecnac     PIC 9(08).
    05  usuario-genero     PIC X.
    05  usuario-direccion  PIC X(50).
    05  usuario-ciudad     PIC X(50).
    05  usuario-telefono1  PIC X(15).
    05  usuario-telefono2  PIC X(15).
    05  usuario-email      PIC X(100).
    05  usuario-feccre     PIC 9(08).
```

- Igual que en el caso anterior, esta ejecución la veremos cuando Desarrollemos el Listado de Clientes.

```
******************************************************************
* Author: Jorge Duarte
* Date: 02-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. BorrarCliente.

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT datos-usuarios
         ASSIGN TO
     "C:\PgmCobols\Data\clientes.dat"
         ORGANIZATION IS INDEXED
         ACCESS MODE IS DYNAMIC
         RECORD KEY IS usuario-rut
         FILE STATUS IS FS-USUARIOS.

 DATA DIVISION.
 FILE SECTION.
 FD  datos-usuarios.
     copy "C:\PgmCobols\FD\FDCliente.cbl".

 WORKING-STORAGE SECTION.
 01  fin-de-archivo      PIC X VALUE 'N'.
 01  opcion              PIC X(1).
 01  continuar           PIC X VALUE 'Y'.
 01  usuario-clave       PIC X(10).
 01  done                PIC X VALUE 'N'.
 01  FS-USUARIOS         PIC XX.
```

NO COPIAR

```
 LINKAGE SECTION.
 01 PAR-INPUT.
     05 RUT-CLIENTE    PIC 9(10).
 01 PAR-OUTPUT.
     05 LK-BORRADO     PIC X.
*       S = Se Borro Cliente
*       N = No Borrado
     05 LK-STATUS      PIC X(02).


 PROCEDURE DIVISION USING PAR-INPUT
                          PAR-OUTPUT.

     OPEN I-O datos-usuarios
     MOVE RUT-CLIENTE   TO usuario-rut
     MOVE "N"       TO LK-BORRADO

     READ datos-usuarios END-READ
     IF FS-USUARIOS = "00" THEN
        DELETE datos-usuarios END-DELETE
        IF FS-USUARIOS = "00" THEN
            MOVE "S"       TO LK-BORRADO
        END-IF
     END-IF

     MOVE FS-USUARIOS  TO LK-STATUS

     CLOSE datos-usuarios
     GOBACK.

 END PROGRAM BorrarCliente.
```

NO COPIAR

Consultar un Registro de un Archivo (ConsultaCliente.CBL)

Para nuestro caso seguimos el ejercicio anterior, sólo recibiremos como campo de entrada el campo KEY del Archivo.

- Archivo C:\PgmCobols\Data\clientes.dat
- Tener la estructura C:\PgmCobols\FD\FDClientes.cbl

```
01  usuario-registro.
    05  usuario-rut        PIC 9(10).
    05  usuario-dv         PIC X.
    05  usuario-nombre     PIC X(30).
    05  usuario-apepat     PIC X(30).
    05  usuario-apemat     PIC X(30).
    05  usuario-fecnac     PIC 9(08).
    05  usuario-genero     PIC X.
    05  usuario-direccion  PIC X(50).
    05  usuario-ciudad     PIC X(50).
    05  usuario-telefono1  PIC X(15).
    05  usuario-telefono2  PIC X(15).
    05  usuario-email      PIC X(100).
    05  usuario-feccre     PIC 9(08).
```

- Igual que en el caso anterior, esta ejecución la veremos cuando Desarrollemos el Listado de Clientes.

```
******************************************************************
* Author: Jorge Duarte
* Date: 02-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ConsultaCliente.

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
    SELECT datos-usuarios
        ASSIGN TO
    "C:\PgmCobols\Data\clientes.dat"
        ORGANIZATION IS INDEXED
        ACCESS MODE IS DYNAMIC
        RECORD KEY IS usuario-rut
        FILE STATUS IS FS-USUARIOS.

 DATA DIVISION.
 FILE SECTION.
 FD  datos-usuarios.
     copy "C:\PgmCobols\FD\FDCliente.cbl".

 WORKING-STORAGE SECTION.
 01  fin-de-archivo      PIC X VALUE 'N'.
 01  opcion              PIC X(1).
 01  continuar           PIC X VALUE 'Y'.
 01  usuario-clave       PIC X(10).
 01  done                PIC X VALUE 'N'.
 01  FS-USUARIOS         PIC XX.

 01 VARIABLE-DE-TRABAJO.
```

NO COPIAR

```
    05 SALIR      PIC X.

01 VARIABLES-PEDIR-PANTALLA.

    05 RUT        PIC 9(10).
    05 DV         PIC X.
    05 NOMBRE     PIC X(30).
    05 APEPAT     PIC X(30).
    05 APEMAT     PIC X(30).
    05 FECNAC     PIC 9(08).
    05 R-FECNAC REDEFINES FECNAC.
       10 DIA-NAC PIC 9(02).
       10 MES-NAC PIC 9(02).
       10 ANO-NAC PIC 9(04).
    05 GENERO     PIC X.
    05 DIRECCION  PIC X(50).
    05 CIUDAD     PIC X(50).
    05 TELEFONO1  PIC X(15).
    05 TELEFONO2  PIC X(15).
    05 EMAIL      PIC X(100).
    05 FECCRE     PIC 9(08).


01 VARIABLES-WS-PANTALLA.

    05 WS-RUT        PIC 9(10).
    05 WS-DV         PIC X.
    05 WS-NOMBRE     PIC X(30).
    05 WS-APEPAT     PIC X(30).
    05 WS-APEMAT     PIC X(30).
    05 WS-FECNAC     PIC 9(08).
    05 WS-R-FECNAC REDEFINES WS-FECNAC.
       10 WS-DIA-NAC PIC 9(02).
       10 WS-MES-NAC PIC 9(02).
       10 WS-ANO-NAC PIC 9(04).
    05 WS-GENERO     PIC X.
    05 WS-DIRECCION  PIC X(50).
    05 WS-CIUDAD     PIC X(50).
    05 WS-TELEFONO1  PIC X(15).
    05 WS-TELEFONO2  PIC X(15).
    05 WS-EMAIL      PIC X(100).
    05 WS-FECCRE     PIC 9(08).

LINKAGE SECTION.
01 PAR-INPUT.
    05 RUT-CLIENTE   PIC 9(10).

SCREEN SECTION.
01  PANTALLA-ENTRADA.
  03  BLANK SCREEN.
  03  LINE 1 COL 30  VALUE "Consulta Cliente".
  03  LINE 5 COL 5  VALUE "RUT..........: ".
  03  LINE 5 COL 22 PIC ZZZZZZZZZ9 FROM RUT .
  03  LINE 5 COL 33  VALUE "-".
  03  LINE 5 COL 35 PIC X(1) FROM DV.
  03  LINE 6 COL 5  VALUE "NOMBRE........: ".
  03  LINE 6 COL 22 PIC X(30) FROM NOMBRE.
  03  LINE 7 COL 5  VALUE "APE. PATERNO..: ".
  03  LINE 7 COL 22 PIC X(30) FROM APEPAT.
  03  LINE 8 COL 5  VALUE "APE. MATERNO..: ".
  03  LINE 8 COL 22 PIC X(30) FROM APEMAT.
  03  LINE 9 COL 5  VALUE "FEC NACIMIENTO: ".
  03  LINE 9 COL 22 PIC 9(02) FROM DIA-NAC.
```

NO COPIAR

```
  03  LINE 9 COL 24 VALUE "/".
  03  LINE 9 COL 25 PIC 9(02) FROM MES-NAC.
  03  LINE 9 COL 27 VALUE "/".
  03  LINE 9 COL 28 PIC 9(04) FROM ANO-NAC.
  03  LINE 10 COL 5 VALUE "GENERO........: ".
  03  LINE 10 COL 22 PIC X FROM GENERO.
  03  LINE 10 COL 25 VALUE "H=HOMBRE; M:MUJER".
  03  LINE 11 COL 5 VALUE "DIRECCION.....: ".
  03  LINE 11 COL 22 PIC X(50) FROM DIRECCION.
  03  LINE 12 COL 5 VALUE "CIUDAD........: ".
  03  LINE 12 COL 22 PIC X(50) FROM CIUDAD.
  03  LINE 13 COL 5 VALUE "TELEFONOS.....: ".
  03  LINE 13 COL 22 PIC X(15) FROM TELEFONO1.
  03  LINE 13 COL 38 PIC X(15) FROM TELEFONO2.
  03  LINE 14 COL 5 VALUE "EMAIL.........: ".
  03  LINE 14 COL 22 PIC X(60) FROM EMAIL.
  03  LINE 22 COL 1 VALUE "-----------------------------------".
  03  LINE 22 COL 35 VALUE "-----------------------------------".
  03  LINE 22 COL 69 VALUE "-------------".
  03  LINE 23 COL 5 VALUE "SALIR (S/N) : ".
  03  LINE 23 COL 19 PIC X USING SALIR.

PROCEDURE DIVISION USING PAR-INPUT.

    OPEN INPUT datos-usuarios
    MOVE RUT-CLIENTE   TO usuario-rut
    READ datos-usuarios END-READ
    IF FS-USUARIOS = "00" THEN
        MOVE usuario-registro TO VARIABLES-PEDIR-PANTALLA
        MOVE "N"   TO SALIR
        PERFORM UNTIL SALIR = 'S' OR 's'

          DISPLAY PANTALLA-ENTRADA
          ACCEPT PANTALLA-ENTRADA

        END-PERFORM
    END-IF
    CLOSE datos-usuarios
    GOBACK.

END PROGRAM ConsultaCliente.
```

NO COPIAR

Mantenedor de un Cliente (Mantenedor.CBL)

Para este caso seguimos usando:

- Archivo C:\PgmCobols\Data\clientes.dat
- Tener la estructura C:\PgmCobols\FD\FDClientes.cbl
- Para Grabar el Registro usaremos la **Rutina GrabarCliente**.
- Para Mostrar algún Error, usaremos **VentanaMsj**.

```
******************************************************************
* Author: Jorge Duarte
* Date: 02-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. Mantenedor.

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

 DATA DIVISION.
 FILE SECTION.


 WORKING-STORAGE SECTION.
 01  fin-de-archivo      PIC X VALUE 'N'.
 01  opcion              PIC X(1).
 01  continuar           PIC X VALUE 'Y'.
 01  usuario-clave       PIC X(10).
 01  done                PIC X VALUE 'N'.
 01  FS-USUARIOS         PIC XX.

 01 VARIABLE-DE-TRABAJO.
     05 FECHA-SYS        PIC 9(08).
     05 CUENTA-1         PIC 9(03).
     05 CUENTA-2         PIC 9(03).
     05 CUENTA-3         PIC 9(03).
     05 WS-ANTES-DE-ARROBA   PIC X(100).
     05 WS-DESPUES-DE-ARROBA PIC X(100).
     05 WS-GRABAR        PIC X.
     05 WS-SALIR         PIC X.

 01 PAR-OUT-GRABAR.
     05  OUT-CODRET-GRABAR   PIC 9(04).
     05  OUT-DESRET-GRABAR   PIC X(50).

 01 VARIABLES-PEDIR-PANTALLA.

     05 RUT         PIC 9(10).
     05 DV          PIC X.
     05 NOMBRE      PIC X(30).
          05 APEPAT      PIC X(30).
          05 APEMAT      PIC X(30).
          05 FECNAC      PIC 9(08).
     05 R-FECNAC REDEFINES FECNAC.
```

NO COPIAR

```
          10 DIA-NAC PIC 9(02).
          10 MES-NAC PIC 9(02).
          10 ANO-NAC PIC 9(04).
            05 GENERO     PIC X.
       05 DIRECCION  PIC X(50).
            05 CIUDAD     PIC X(50).
            05 TELEFONO1  PIC X(15).
            05 TELEFONO2  PIC X(15).
            05 EMAIL      PIC X(100).
            05 FECCRE     PIC 9(08).
       05 SALIR      PIC X.
       05 MSG-ERROR  PIC X(70).

    01 PARAMETROS-VALIDA-FECHA.
       05 PAR-INP-FECHA.
           10 FECHA-X     PIC X(08).
           10 FORMATO-X   PIC X VALUE "2".
       05 PAR-OUT-FECHA.
           10 FECHA-VALIDA PIC X.

    01 PARAMETROS-LARGO-STRING.
       05 STRING-CALCULAR   PIC X(500).
       05 STRING-LARGO      PIC 9(03).

    01 VARIABLES-WS-PANTALLA.

       05 WS-RUT        PIC 9(10).
       05 WS-DV         PIC X.
       05 WS-NOMBRE     PIC X(30).
       05 WS-APEPAT     PIC X(30).
       05 WS-APEMAT     PIC X(30).
       05 WS-FECNAC     PIC 9(08).
       05 WS-R-FECNAC REDEFINES WS-FECNAC.
          10 WS-DIA-NAC PIC 9(02).
          10 WS-MES-NAC PIC 9(02).
          10 WS-ANO-NAC PIC 9(04).
       05 WS-GENERO     PIC X.
       05 WS-DIRECCION  PIC X(50).
       05 WS-CIUDAD     PIC X(50).
       05 WS-TELEFONO1  PIC X(15).
       05 WS-TELEFONO2  PIC X(15).
       05 WS-EMAIL      PIC X(100).
       05 WS-FECCRE     PIC 9(08).



    01 PAR-INP.
       10 INP-RUT    PIC 9(10).
       10 INP-DV     PIC X.
    01 PAR-OUT.
       10 OUT-CODRET PIC 9.

    SCREEN SECTION.
    01  PANTALLA-ENTRADA.
      03  BLANK SCREEN.
      03  LINE 1 COL 30  VALUE "Mantenedor Clientes".
    *  03  LINE 1 COL 70 PIC X(10) USING FECHA-SYS DISPLAY.
      03  LINE 5 COL 5  VALUE "RUT...........: ".
      03  LINE 5 COL 22 PIC ZZZZZZZZZ9 USING RUT .
      03  LINE 5 COL 33  VALUE "-".
      03  LINE 5 COL 35 PIC X(1) USING DV.
      03  LINE 6 COL 5  VALUE "NOMBRE........: ".
      03  LINE 6 COL 22 PIC X(30) USING NOMBRE.
```

NO COPIAR

```
           03  LINE 7 COL 5  VALUE "APE. PATERNO..: ".
           03  LINE 7 COL 22 PIC X(30) USING APEPAT.
           03  LINE 8 COL 5  VALUE "APE. MATERNO..: ".
           03  LINE 8 COL 22 PIC X(30) USING APEMAT.
           03  LINE 9 COL 5  VALUE "FEC NACIMIENTO: ".
           03  LINE 9 COL 22 PIC 9(02) USING DIA-NAC.
           03  LINE 9 COL 24 VALUE "/".
           03  LINE 9 COL 25 PIC 9(02) USING MES-NAC.
           03  LINE 9 COL 27 VALUE "/".
           03  LINE 9 COL 28 PIC 9(04) USING ANO-NAC.
           03  LINE 10 COL 5 VALUE "GENERO........: ".
           03  LINE 10 COL 22 PIC X USING GENERO.
           03  LINE 10 COL 25 VALUE "H=HOMBRE; M:MUJER".
           03  LINE 11 COL 5 VALUE "DIRECCION.....: ".
           03  LINE 11 COL 22 PIC X(50) USING DIRECCION.
           03  LINE 12 COL 5 VALUE "CIUDAD........: ".
           03  LINE 12 COL 22 PIC X(50) USING CIUDAD.
           03  LINE 13 COL 5 VALUE "TELEFONOS.....: ".
           03  LINE 13 COL 22 PIC X(15) USING TELEFONO1.
           03  LINE 13 COL 38 PIC X(15) USING TELEFONO2.
           03  LINE 14 COL 5 VALUE "EMAIL........: ".
           03  LINE 14 COL 22 PIC X(60) USING EMAIL.
           03  LINE 22 COL 1 VALUE "----------------------------------".
           03  LINE 22 COL 35 VALUE "----------------------------------".
           03  LINE 22 COL 69 VALUE "-------------".
           03  LINE 23 COL 5 VALUE "Intro-Validar".
           03  LINE 23 COL 30 VALUE "SALIR (S/N) : ".
           03  LINE 23 COL 44 PIC X USING SALIR.
       *   03  LINE 24 COL 1 PIC X(70) USING MSG-ERROR .


        PROCEDURE DIVISION.

            ACCEPT FECHA-SYS FROM DATE
            DISPLAY FECHA-SYS AT LINE 1 COLUMN 70
            MOVE "N"   TO SALIR
            PERFORM UNTIL SALIR = 'S' OR 's'
              DISPLAY PANTALLA-ENTRADA
              ACCEPT PANTALLA-ENTRADA
              MOVE SPACES   TO MSG-ERROR
              IF SALIR = "N" THEN
                MOVE VARIABLES-PEDIR-PANTALLA  TO
                    VARIABLES-WS-PANTALLA

                PERFORM VALIDAR-DATOS
                IF MSG-ERROR = SPACES THEN
                  PERFORM PEDIR-GRABAR
                END-IF

              END-IF
            END-PERFORM
            STOP RUN.

        VALIDAR-DATOS.
            MOVE SPACES   TO MSG-ERROR
            PERFORM VALIDA-RUT
            PERFORM VALIDA-NOMBRE-APELLIDOS
            PERFORM VALIDA-FECHA-NACIMIENTO
            PERFORM VALIDA-GENERO
            PERFORM VALIDA-DIRECCION
            PERFORM VALIDA-CIUDAD
            PERFORM VALIDA-TELEFONOS
            PERFORM VALIDA-EMAIL
```

NO COPIAR

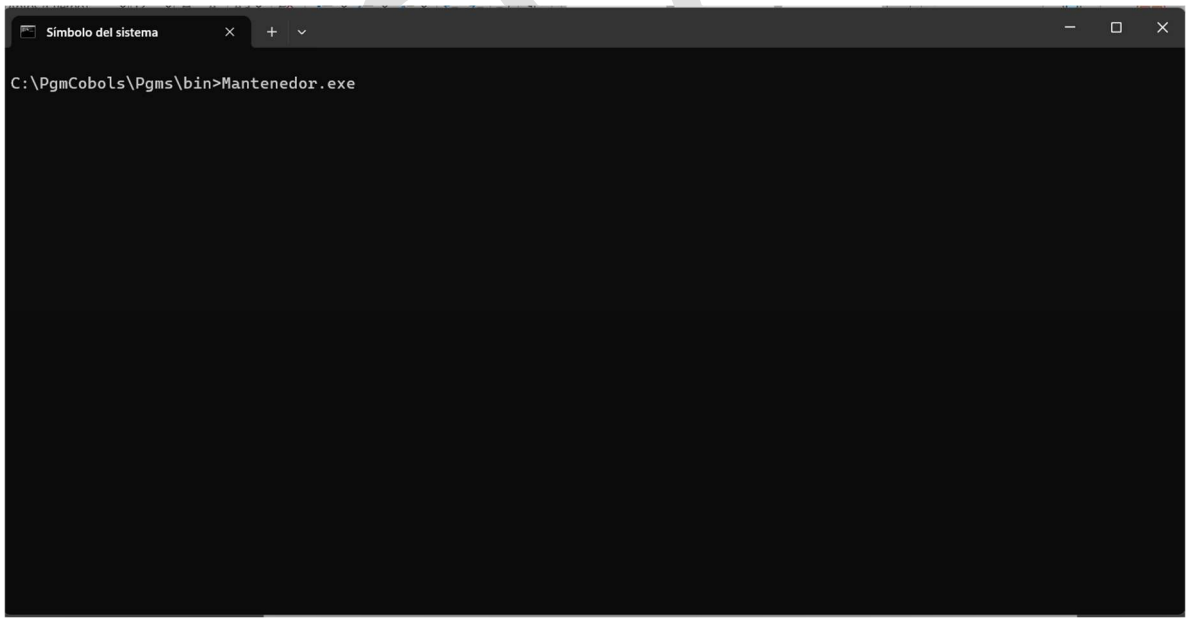
```
    IF MSG-ERROR NOT = SPACES THEN
       CALL "VentanaMsj" USING MSG-ERROR
    END-IF
    .
VALIDA-RUT.
    IF WS-RUT = 0 OR WS-DV = SPACES THEN
        MOVE "RUT INCORRECTO" TO MSG-ERROR
    ELSE
        MOVE WS-RUT        TO INP-RUT
        MOVE WS-DV         TO INP-DV
        MOVE 0          TO OUT-CODRET
        CALL "ValidaRut" USING PAR-INP PAR-OUT
        IF OUT-CODRET = 1 THEN
           MOVE "RUT INCORRECTO" TO MSG-ERROR
        END-IF
    END-IF
    .
VALIDA-NOMBRE-APELLIDOS.
    IF MSG-ERROR = SPACES
        IF WS-NOMBRE = SPACES OR
           WS-APEPAT = SPACES OR
           WS-APEMAT = SPACES THEN
           MOVE "NOMBRE O APELLIDOS INCORRECTOS"
                                TO MSG-ERROR
        END-IF
    END-IF
    .

VALIDA-FECHA-NACIMIENTO.
    IF MSG-ERROR = SPACES
       MOVE WS-FECNAC     TO FECHA-X
       MOVE " "           TO FECHA-VALIDA
       CALL "ValidarFecha" using PAR-INP-FECHA PAR-OUT-FECHA

       IF FECHA-VALIDA = "N" THEN
           MOVE "FECHA NACIMIENTO INCORRECTA"
                                TO MSG-ERROR
       END-IF
    END-IF
    .

VALIDA-GENERO.
    IF MSG-ERROR = SPACES
       IF WS-GENERO = SPACES OR
          (WS-GENERO NOT = "H" AND WS-GENERO NOT = "M") THEN

           MOVE "Género debe ser H ó M"
                                TO MSG-ERROR
       END-IF
    END-IF
    .

VALIDA-DIRECCION.
    IF MSG-ERROR = SPACES

       MOVE WS-DIRECCION   TO STRING-CALCULAR
       PERFORM CALCULAR-LARGO-STRING
       IF STRING-LARGO < 5 THEN
          MOVE "DEBE INGRESAR UNA DIRECCION MINIMO 5 CARACTERES"
                                TO MSG-ERROR
       END-IF
    END-IF
```

NO COPIAR

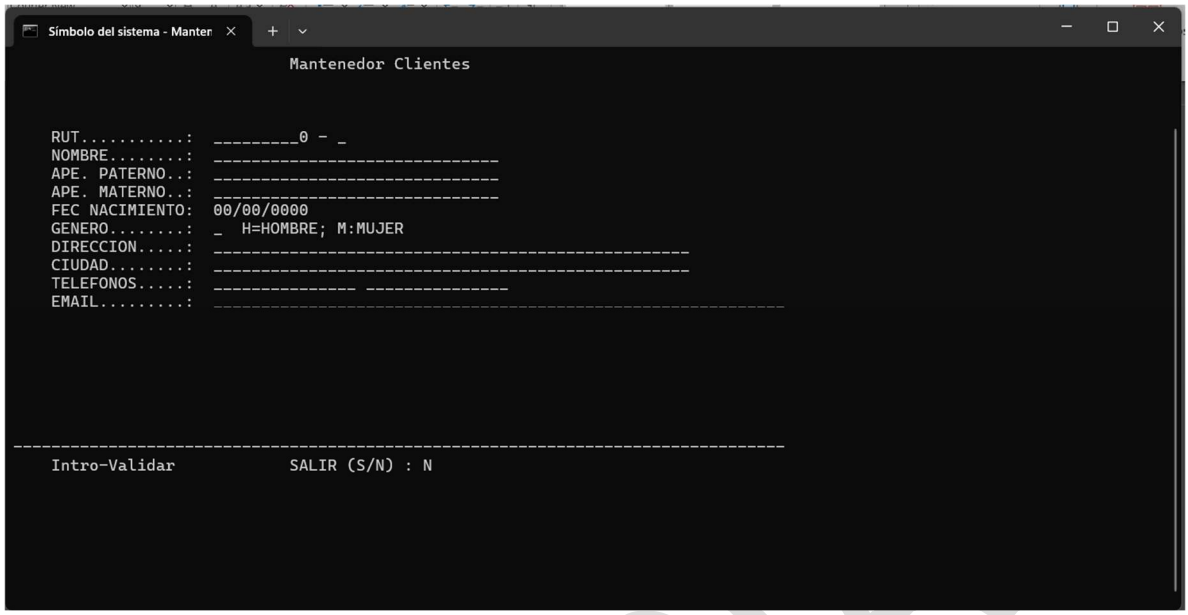
```
    .

VALIDA-CIUDAD.
    IF MSG-ERROR = SPACES
        MOVE WS-CIUDAD     TO STRING-CALCULAR
        PERFORM CALCULAR-LARGO-STRING
       IF STRING-LARGO < 5 THEN
          MOVE "DEBE INGRESAR UNA CIUDAD MINIMO 5 CARACTERES"
                                TO MSG-ERROR
       END-IF
    END-IF
    .

VALIDA-TELEFONOS.
    IF MSG-ERROR = SPACES
       IF WS-TELEFONO1 = SPACES AND
          WS-TELEFONO2 = SPACES
          MOVE "DEBE INGRESAR AL MENOS 1 TELEFONO"
                                TO MSG-ERROR
       END-IF
    END-IF
    .

VALIDA-EMAIL.
    IF MSG-ERROR = SPACES
       IF WS-EMAIL = SPACES
          MOVE "DEBE INGRESAR EMAIL (1)"
                                TO MSG-ERROR
       ELSE
           INSPECT WS-EMAIL TALLYING CUENTA-1 FOR ALL "@"

           IF CUENTA-1 = 1 THEN
               UNSTRING WS-EMAIL DELIMITED BY "@"
               INTO     WS-ANTES-DE-ARROBA,
                        WS-DESPUES-DE-ARROBA
               IF WS-ANTES-DE-ARROBA(1:3) NOT = SPACES THEN
                  INSPECT WS-DESPUES-DE-ARROBA TALLYING
                  CUENTA-3 FOR ALL "."
                  IF CUENTA-3 = 0 THEN
                     MOVE
                "DEBE INGRESAR EMAIL CON FORMATO CORRECTO (2)"
                                TO MSG-ERROR
                  ELSE
                     IF WS-DESPUES-DE-ARROBA(1:3) = SPACES THEN
                        MOVE
                      "DEBE INGRESAR EMAIL CON FORMATO CORRECTO (3)"
                        TO MSG-ERROR
                     END-IF
                  END-IF
               ELSE
             MOVE "DEBE INGRESAR EMAIL CON FORMATO CORRECTO (4)"
                                TO MSG-ERROR
               END-IF

           ELSE
              MOVE "DEBE INGRESAR EMAIL CON FORMATO CORRECTO (5)"
                                TO MSG-ERROR
           END-IF

       END-IF
    END-IF
    .
```

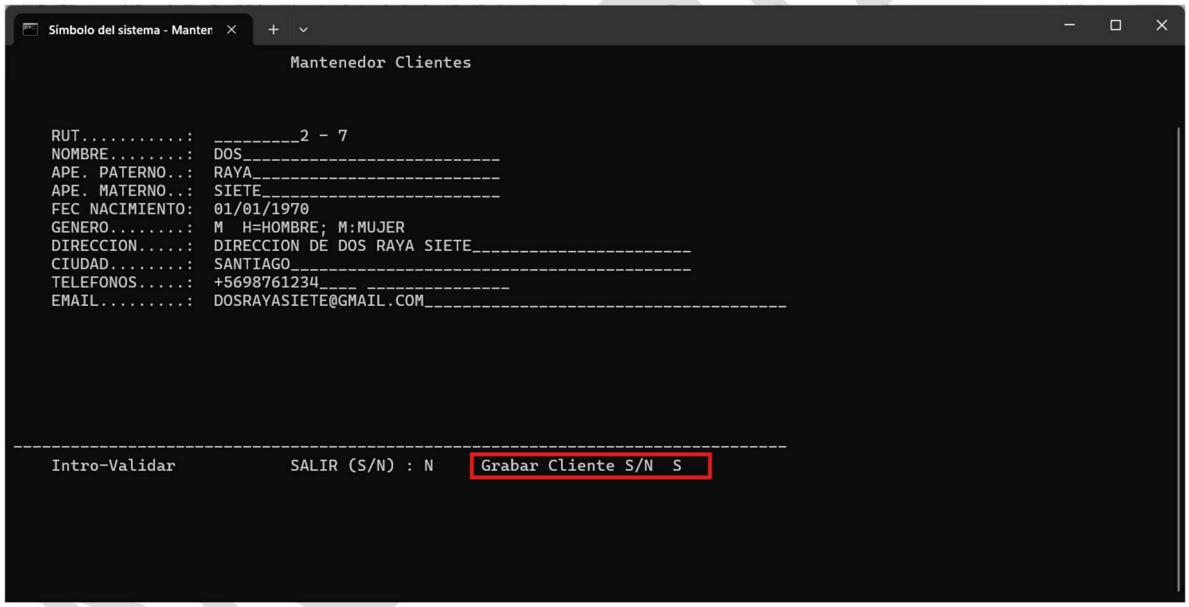
NO COPIAR

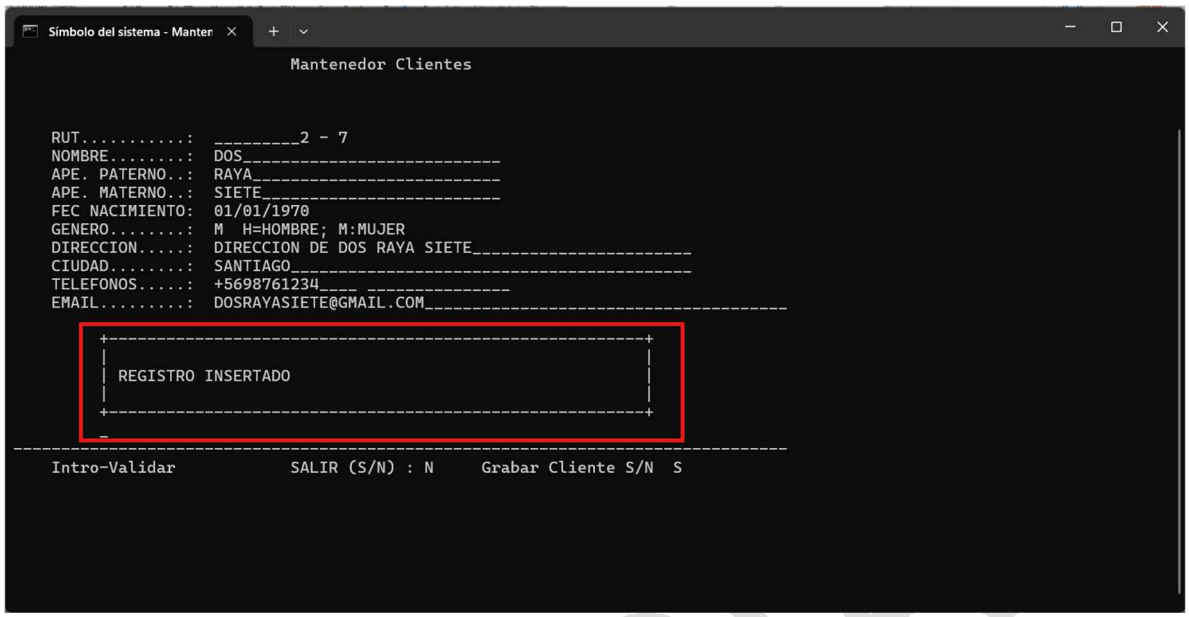
```
CALCULAR-LARGO-STRING.

    MOVE 0             TO STRING-LARGO
    CALL "LargoString" USING STRING-CALCULAR STRING-LARGO
    .

PEDIR-GRABAR.

    MOVE " "   TO WS-GRABAR
    PERFORM UNTIL WS-GRABAR = "S" OR "N"
       DISPLAY "Grabar Cliente S/N" AT LINE 23 COLUMN 50
       ACCEPT WS-GRABAR             AT LINE 23 COLUMN 70

       IF WS-GRABAR = "S" OR "s" THEN
           MOVE 0               TO OUT-CODRET-GRABAR
           MOVE SPACES          TO OUT-DESRET-GRABAR
           CALL "GrabarCliente" USING VARIABLES-WS-PANTALLA
                                      PAR-OUT-GRABAR

           MOVE OUT-DESRET-GRABAR TO MSG-ERROR
           CALL "VentanaErr" USING MSG-ERROR

           IF OUT-CODRET-GRABAR = 0 THEN
              MOVE SPACES           TO MSG-ERROR
              MOVE "S"              TO SALIR
           END-IF
       END-IF
    END-PERFORM
    .

END PROGRAM Mantenedor.
```



COPIAR


Símbolo del sistema - Manter
Mantenedor Clientes
RUT.....: _____0 - _
NOMBRE.....: _____
APE. PATERNO..: _____
APE. MATERNO..: _____
FEC NACIMIENTO: 00/00/0000
GENERO.....: _ H=HOMBRE; M:MUJER
DIRECCION.....: _____
CIUDAD.....: _____
TELEFONOS.....: _____ _____
EMAIL.....: _____
Intro-Validar
SALIR (S/N) : N


Símbolo del sistema - Manter
Mantenedor Clientes
RUT.....: _____2 - 7
NOMBRE.....: DOS_____
APE. PATERNO..: RAYA_____
APE. MATERNO..: SIETE_____
FEC NACIMIENTO: 01/01/1970
GENERO.....: M H=HOMBRE; M:MUJER
DIRECCION.....: DIRECCION DE DOS RAYA SIETE_____
CIUDAD.....: SANTIAGO_____
TELEFONOS.....: +5698761234____ _____
EMAIL.....: DOSRAYASIETE@GMAIL.COM_____
Intro-Validar
SALIR (S/N) : N
Grabar Cliente S/N S


Símbolo del sistema - Manter
Mantenedor Clientes
RUT.....: _____2 - 7
NOMBRE.....: DOS_____
APE. PATERNO..: RAYA_____
APE. MATERNO..: SIETE_____
FEC NACIMIENTO: 01/01/1970
GENERO.....: M H=HOMBRE; M:MUJER
DIRECCION.....: DIRECCION DE DOS RAYA SIETE_____
CIUDAD.....: SANTIAGO_____
TELEFONOS.....: +5698761234____ _____
EMAIL.....: DOSRAYASIETE@GMAIL.COM_____
REGISTRO INSERTADO
Intro-Validar
SALIR (S/N) : N
Grabar Cliente S/N S

NO COPIA

Listado de Registros 1 (ListadoClientes.CBL)

El presente programa muestra una forma de mostrar registros, en este caso de Clientes con un posicionamiento por Rut, esto puede ser modificado de acuerdo a las necesidades que cada persona requiera.

```
******************************************************************
* Author: Jorge Duarte
* Date: 02-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ListadoClientes.

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT datos-usuarios
         ASSIGN TO
     "C:\PgmCobols\Data\clientes.dat"
         ORGANIZATION IS INDEXED
         ACCESS MODE IS DYNAMIC
         RECORD KEY IS usuario-rut
         FILE STATUS IS FS-USUARIOS.

 DATA DIVISION.
 FILE SECTION.
 FD  datos-usuarios.
     copy "C:\PgmCobols\FD\FDCliente.cbl".

 WORKING-STORAGE SECTION.
 01  fin-de-archivo      PIC X VALUE 'N'.
 01  opcion              PIC X(1).
 01  continuar           PIC X VALUE 'Y'.
 01  usuario-clave       PIC X(10).
 01  done                PIC X VALUE 'N'.
 01  FS-USUARIOS         PIC XX.
 01  SALIR               PIC X.

 01 VARIABLES-DE-TRABAJO.
     05 BLANCO           PIC X.
     05 RUT-POS          PIC 9(10).
     05 WS-NOMBRE        PIC X(50).
     05 POS-ARR          PIC 9(3).
     05 WS-LINEA         PIC 9(3).
     05 WS-LIN           PIC 9(2).
     05 WS-OPC           PIC X.
     05 ARR-DETALLE OCCURS 15 TIMES.
        10 ARR-LIN       PIC 9(3).
        10 ARR-RUT       PIC 9(10).
        10 ARR-GUI       PIC X.
        10 ARR-DV        PIC X.
        10 ARR-NOM       PIC X(50).

 01 PAR-OUT-BORRAR.
     05 LK-BORRADO    PIC X.
*       S = Se Borro Cliente
*       N = No Borrado
     05 LK-STATUS     PIC X(02).
```

NO COPIAR

```
SCREEN SECTION.
01  PANTALLA-ENTRADA.
  03  BLANK SCREEN.
  03  LINE 1 COL 30  VALUE "Listado Clientes".
  03  LINE 3 COL 5   VALUE "Posicionar Rut : ".
  03  LINE 3 COL 24 PIC 9(10) USING RUT-POS.

01 PANTALLA-VISUALIZACION.
  03  LINE 5 COL 01 VALUE "====================================".
  03  LINE 5 COL 37 VALUE "====================================".
  03  LINE 5 COL 73 VALUE "=====".
  03  LINE 6 COL 01 VALUE "LIN RUT           NOMBRE".
  03  LINE 7 COL 01 VALUE "====================================".
  03  LINE 7 COL 37 VALUE "====================================".
  03  LINE 7 COL 73 VALUE "=====".

  03  LINE 8 COL 1  PIC ZZZ FROM ARR-LIN(1).
  03  LINE 8 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(1).
  03  LINE 8 COL 15 PIC X FROM ARR-GUI(1).
  03  LINE 8 COL 16 PIC X  FROM ARR-DV(1).
  03  LINE 8 COL 19 PIC X(50) FROM ARR-NOM(1).

  03  LINE 9 COL 1  PIC ZZZ FROM ARR-LIN(2).
  03  LINE 9 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(2).
  03  LINE 9 COL 15 PIC X FROM ARR-GUI(2).
  03  LINE 9 COL 16 PIC X  FROM ARR-DV(2).
  03  LINE 9 COL 19 PIC X(50) FROM ARR-NOM(2).

  03  LINE 10 COL 1  PIC ZZZ FROM ARR-LIN(3).
  03  LINE 10 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(3).
  03  LINE 10 COL 15 PIC X FROM ARR-GUI(3).
  03  LINE 10 COL 16 PIC X  FROM ARR-DV(3).
  03  LINE 10 COL 19 PIC X(50) FROM ARR-NOM(3).


  03  LINE 11 COL 1  PIC ZZZ FROM ARR-LIN(4).
  03  LINE 11 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(4).
  03  LINE 11 COL 15 PIC X FROM ARR-GUI(4).
  03  LINE 11 COL 16 PIC X  FROM ARR-DV(4).
  03  LINE 11 COL 19 PIC X(50) FROM ARR-NOM(4).

  03  LINE 12 COL 1  PIC ZZZ FROM ARR-LIN(5).
  03  LINE 12 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(5).
  03  LINE 12 COL 15 PIC X FROM ARR-GUI(5).
  03  LINE 12 COL 16 PIC X  FROM ARR-DV(5).
  03  LINE 12 COL 19 PIC X(50) FROM ARR-NOM(5).

  03  LINE 13 COL 1  PIC ZZZ FROM ARR-LIN(6).
  03  LINE 13 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(6).
  03  LINE 13 COL 15 PIC X FROM ARR-GUI(6).
  03  LINE 13 COL 16 PIC X  FROM ARR-DV(6).
  03  LINE 13 COL 19 PIC X(50) FROM ARR-NOM(6).

  03  LINE 14 COL 1  PIC ZZZ FROM ARR-LIN(7).
  03  LINE 14 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(7).
  03  LINE 14 COL 15 PIC X FROM ARR-GUI(7).
  03  LINE 14 COL 16 PIC X  FROM ARR-DV(7).
  03  LINE 14 COL 19 PIC X(50) FROM ARR-NOM(7).

  03  LINE 15 COL 1  PIC ZZZ FROM ARR-LIN(8).
  03  LINE 15 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(8).
  03  LINE 15 COL 15 PIC X FROM ARR-GUI(8).
  03  LINE 15 COL 16 PIC X  FROM ARR-DV(8).
```

NO COPIAR

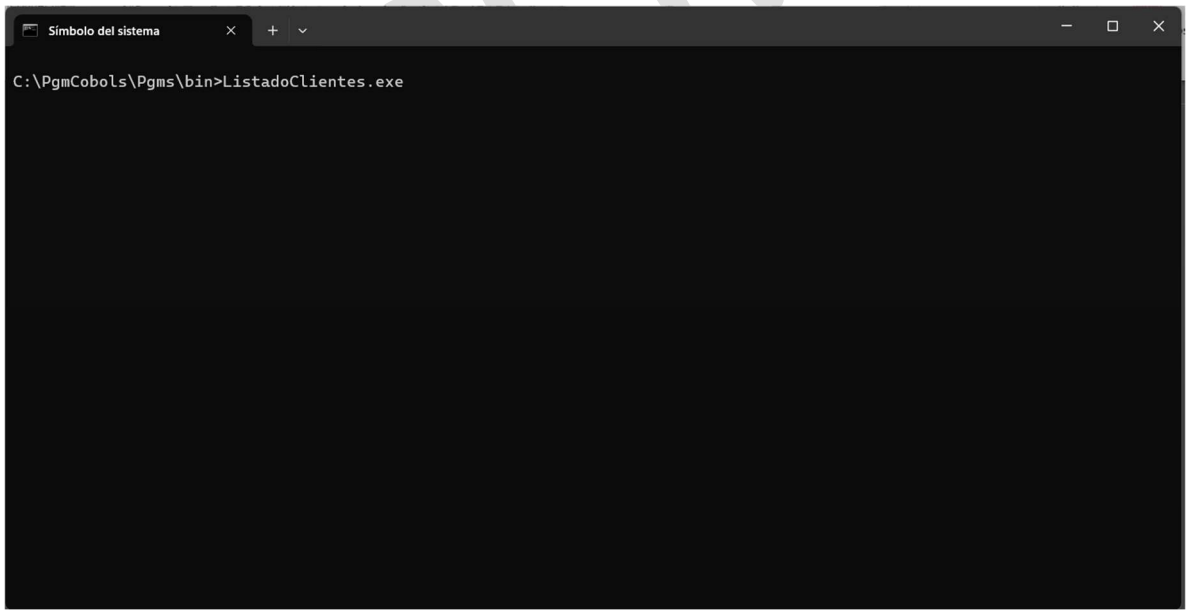
```
  03  LINE 15 COL 19 PIC X(50) FROM ARR-NOM(8).

  03  LINE 16 COL 1  PIC ZZZ FROM ARR-LIN(9).
  03  LINE 16 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(9).
  03  LINE 16 COL 15 PIC X FROM ARR-GUI(9).
  03  LINE 16 COL 16 PIC X  FROM ARR-DV(9).
  03  LINE 16 COL 19 PIC X(50) FROM ARR-NOM(9).

  03  LINE 17 COL 1  PIC ZZZ FROM ARR-LIN(10).
  03  LINE 17 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(10).
  03  LINE 17 COL 15 PIC X FROM ARR-GUI(10).
  03  LINE 17 COL 16 PIC X  FROM ARR-DV(10).
  03  LINE 17 COL 19 PIC X(50) FROM ARR-NOM(10).

  03  LINE 18 COL 1  PIC ZZZ FROM ARR-LIN(11).
  03  LINE 18 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(11).
  03  LINE 18 COL 15 PIC X FROM ARR-GUI(11).
  03  LINE 18 COL 16 PIC X  FROM ARR-DV(11).
  03  LINE 18 COL 19 PIC X(50) FROM ARR-NOM(11).

  03  LINE 19 COL 1  PIC ZZZ FROM ARR-LIN(12).
  03  LINE 19 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(12).
  03  LINE 19 COL 15 PIC X FROM ARR-GUI(12).
  03  LINE 19 COL 16 PIC X  FROM ARR-DV(12).
  03  LINE 19 COL 19 PIC X(50) FROM ARR-NOM(12).

  03  LINE 20 COL 1  PIC ZZZ FROM ARR-LIN(13).
  03  LINE 20 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(13).
  03  LINE 20 COL 15 PIC X FROM ARR-GUI(13).
  03  LINE 20 COL 16 PIC X  FROM ARR-DV(13).
  03  LINE 20 COL 19 PIC X(50) FROM ARR-NOM(13).

  03  LINE 21 COL 1  PIC ZZZ FROM ARR-LIN(14).
  03  LINE 21 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(14).
  03  LINE 21 COL 15 PIC X FROM ARR-GUI(14).
  03  LINE 21 COL 16 PIC X  FROM ARR-DV(14).
  03  LINE 21 COL 19 PIC X(50) FROM ARR-NOM(14).

  03  LINE 22 COL 1  PIC ZZZ FROM ARR-LIN(15).
  03  LINE 22 COL 5  PIC ZZZZZZZZZZ FROM ARR-RUT(15).
  03  LINE 22 COL 15 PIC X FROM ARR-GUI(15).
  03  LINE 22 COL 16 PIC X  FROM ARR-DV(15).
  03  LINE 22 COL 19 PIC X(50) FROM ARR-NOM(15).

01 PANTALLA-SALIR.
  03 LINE 24 COL 5 VALUE "SALIR (S/N) : ".
  03 LINE 24 COL 19 PIC X USING SALIR.
  03 LINE 24 COL 21 VALUE "LINEA :".
  03 LINE 24 COL 28 PIC 9(02) USING WS-LIN.
  03 LINE 24 COL 31 VALUE "Opcion C=Consultar; B=Borrar :".
  03 LINE 24 COL 64 PIC X USING WS-OPC.

PROCEDURE DIVISION.
INICIO.
    OPEN INPUT datos-usuarios
    MOVE 0      TO RUT-POS
    MOVE "N"    TO SALIR
    PERFORM UNTIL SALIR = "S" OR "s"
        DISPLAY PANTALLA-ENTRADA
        ACCEPT  PANTALLA-ENTRADA

        PERFORM mostrar-todos-los-usuarios
        PERFORM PEDIR-SALIR
```

NO COPIAR

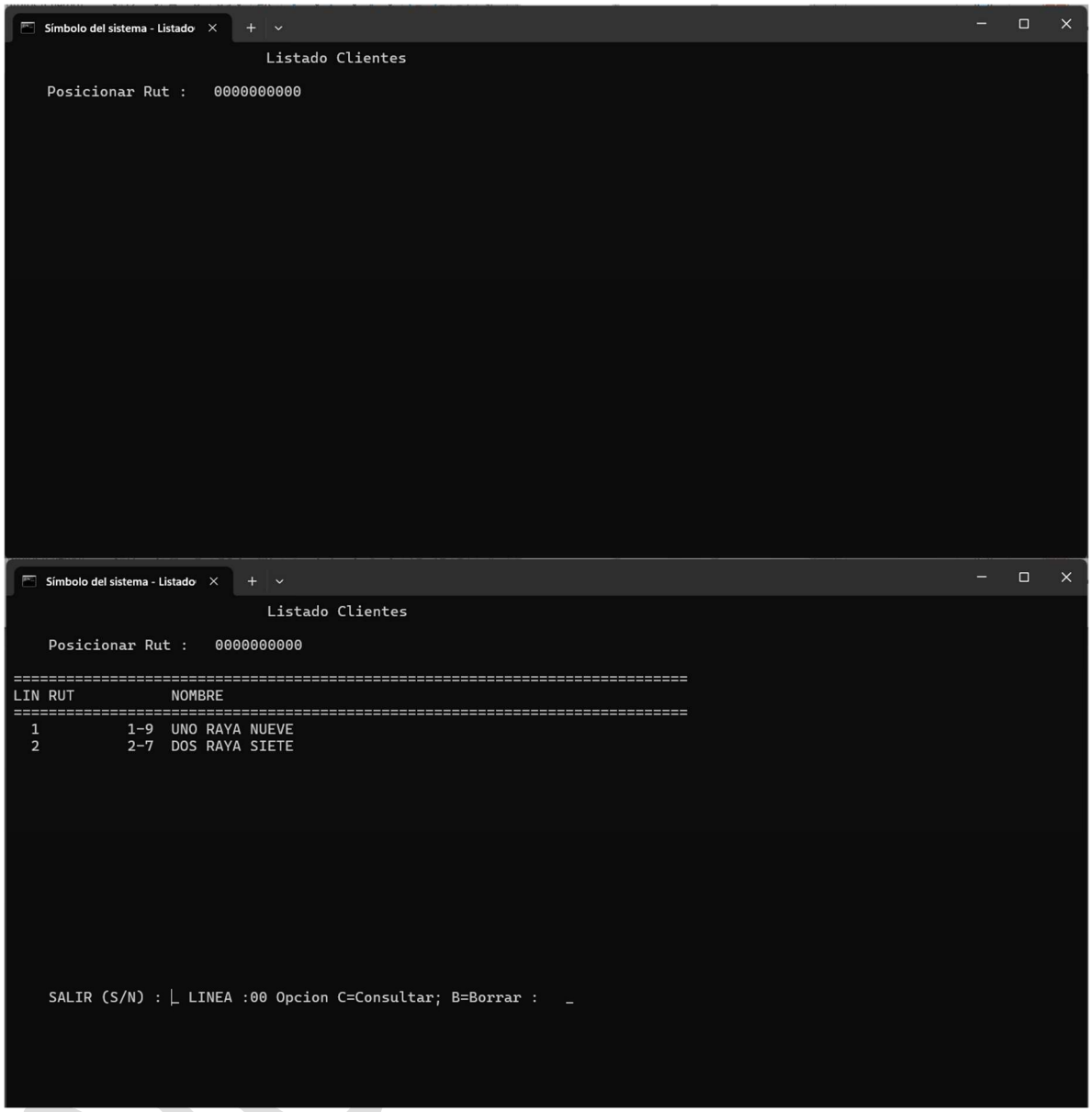
```
    END-PERFORM
    CLOSE datos-usuarios
    STOP RUN.

mostrar-todos-los-usuarios.
    PERFORM cargar-usuarios

    DISPLAY PANTALLA-ENTRADA
    ACCEPT  PANTALLA-VISUALIZACION

     .

cargar-usuarios.
    PERFORM START-READING

    PERFORM VARYING POS-ARR FROM 1 BY 1 UNTIL POS-ARR > 15
       MOVE ZEROES   TO ARR-LIN(POS-ARR)
                        ARR-RUT(POS-ARR)
       MOVE SPACES   TO ARR-DV (POS-ARR)
                        ARR-GUI(POS-ARR)
                        ARR-NOM(POS-ARR)
    END-PERFORM

    MOVE 1  TO POS-ARR

    PERFORM UNTIL fin-de-archivo = 'Y' OR POS-ARR > 15
        READ datos-usuarios
            AT END
                MOVE 'Y' TO  fin-de-archivo
            NOT AT END

                MOVE POS-ARR         TO ARR-LIN    (POS-ARR)
                MOVE usuario-rut     TO ARR-RUT    (POS-ARR)
                MOVE "-"             TO ARR-GUI    (POS-ARR)
                MOVE usuario-dv      TO ARR-DV     (POS-ARR)
                STRING usuario-nombre DELIMITED BY " "
                       " "            DELIMITED BY SIZE
                       usuario-apepat DELIMITED BY " "
                       " "            DELIMITED BY SIZE
                       usuario-apemat DELIMITED BY " "
                                   INTO ARR-NOM    (POS-ARR)

                ADD  1               TO POS-ARR
        END-READ
    END-PERFORM

    .

START-READING.
    MOVE 'N'        TO fin-de-archivo
    MOVE RUT-POS    TO usuario-rut
    START datos-usuarios KEY IS NOT < usuario-rut
        INVALID KEY
            DISPLAY "Error al iniciar lectura de registros."
            MOVE "Y"    TO fin-de-archivo
    END-START.

PEDIR-SALIR.
    MOVE " "            TO SALIR
    PERFORM UNTIL SALIR = "S" OR "s" or "N" or "n"
       DISPLAY PANTALLA-SALIR
       ACCEPT  PANTALLA-SALIR
       IF WS-LIN NOT = 0 AND WS-LIN < 16 THEN
```

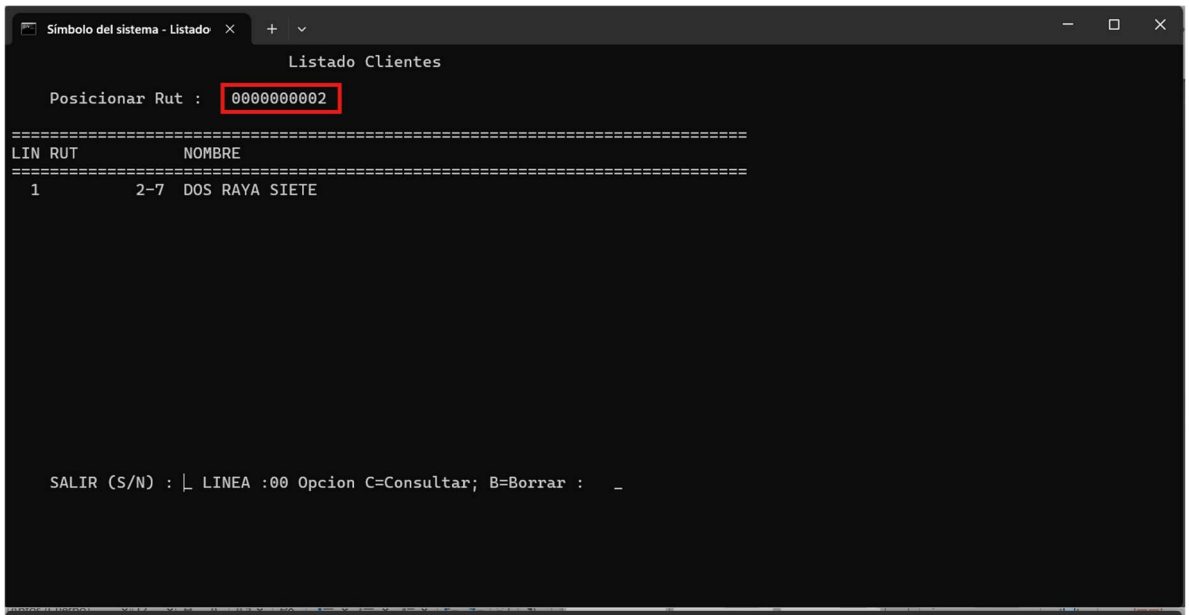
NO COPIAR

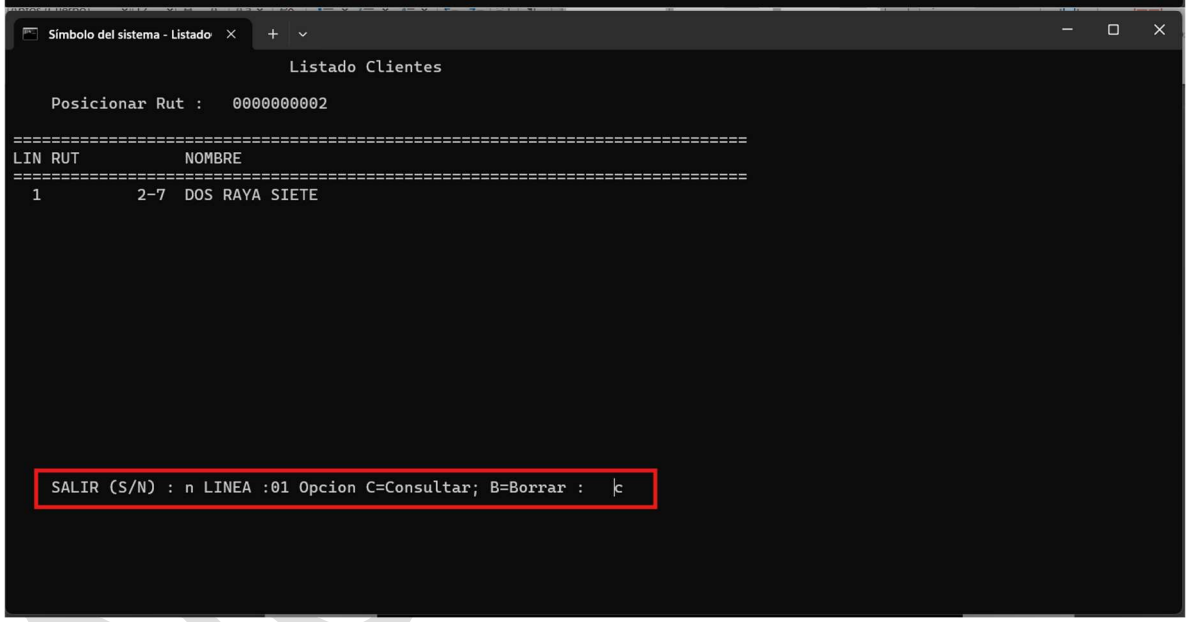
```
            IF WS-OPC = "C" OR "c" THEN
               PERFORM LLAMA-CONSULTA
               MOVE ZEROES  TO WS-LIN
               MOVE SPACES  TO WS-OPC
               MOVE "N" TO SALIR
            END-IF
            IF WS-OPC = "B" OR "b" THEN
               PERFORM ELIMINA-CLIENTE
               MOVE ZEROES  TO WS-LIN
               MOVE SPACES  TO WS-OPC
               MOVE "N" TO SALIR
            END-IF
         END-IF
      END-PERFORM
      .

   LLAMA-CONSULTA.
      IF ARR-RUT(WS-LIN) NOT = 0 THEN
          CALL "ConsultaCliente" USING ARR-RUT(WS-LIN)
      END-IF
      .

    ELIMINA-CLIENTE.
      IF ARR-RUT(WS-LIN) NOT = 0 THEN
          CALL "BorrarCliente" USING ARR-RUT(WS-LIN)
                                     PAR-OUT-BORRAR
      END-IF
      .

   END PROGRAM ListadoClientes.
```

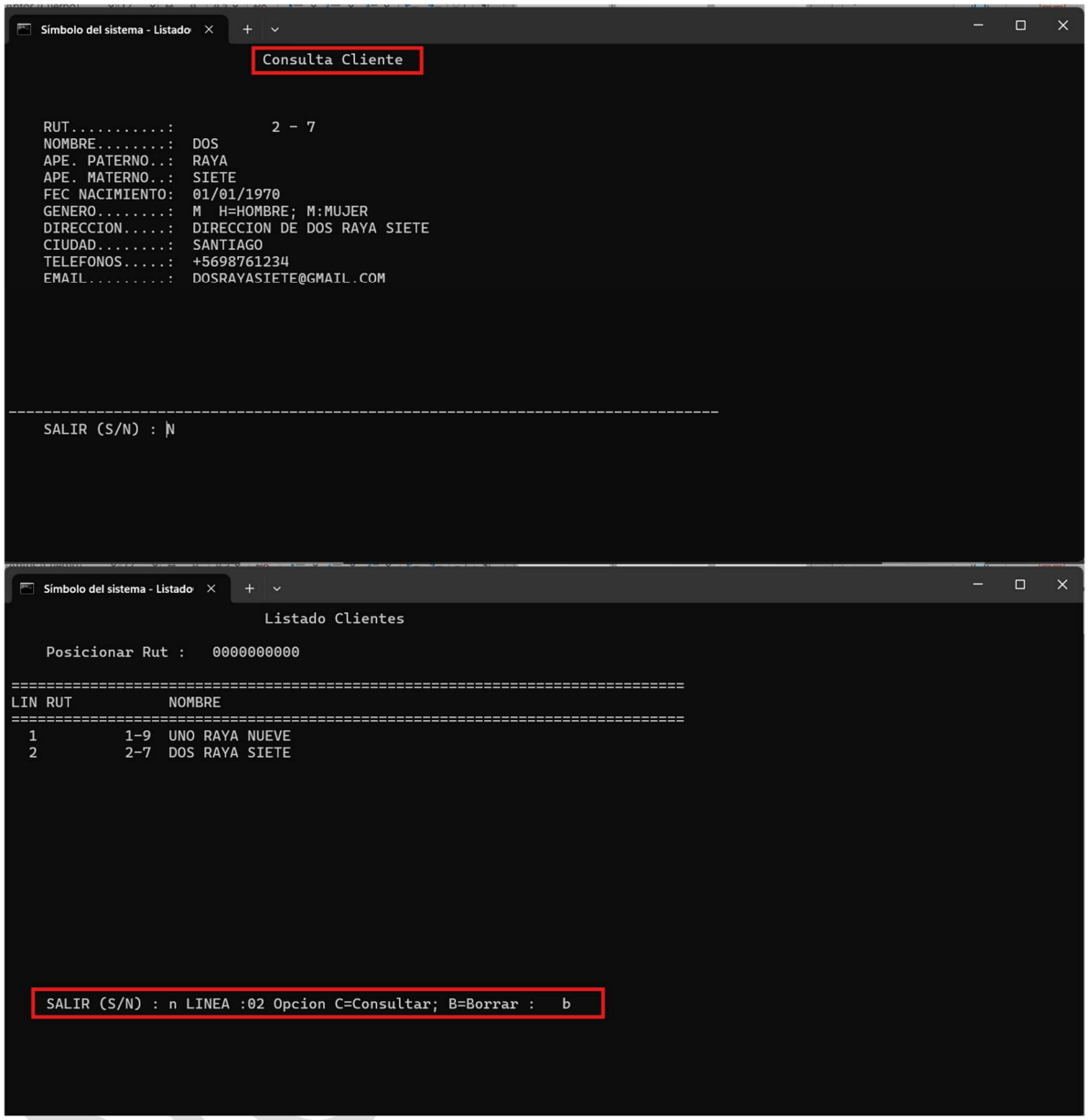


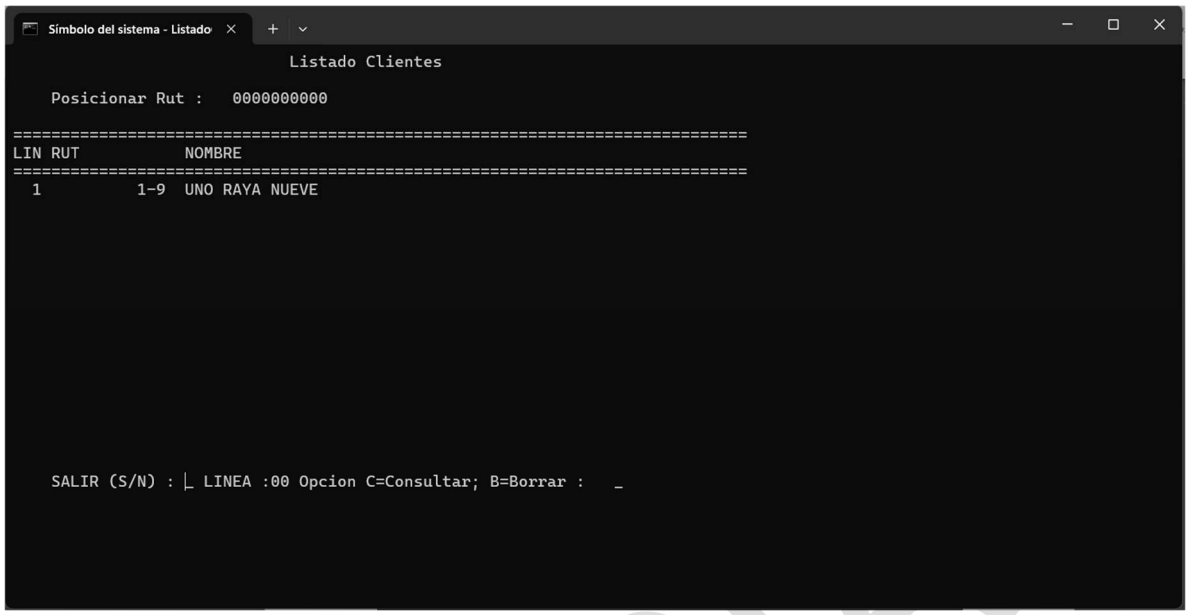
COPIAR


Símbolo del sistema - Listado
Listado Clientes
Posicionar Rut : 0000000000
Símbolo del sistema - Listado
Listado Clientes
Posicionar Rut : 0000000000
LIN RUT NOMBRE
1 1-9 UNO RAYA NUEVE
2 2-7 DOS RAYA SIETE
SALIR (S/N) : LINEA :00 Opcion C=Consultar; B=Borrar :


Símbolo del sistema - Listado
Listado Clientes
Posicionar Rut : 0000000002
LIN RUT NOMBRE
1 2-7 DOS RAYA SIETE
SALIR (S/N) : LINEA :00 Opcion C=Consultar; B=Borrar :


Símbolo del sistema - Listado
Listado Clientes
Posicionar Rut : 0000000002
LIN RUT NOMBRE
1 2-7 DOS RAYA SIETE
SALIR (S/N) : n LINEA :01 Opcion C=Consultar; B=Borrar : c


Símbolo del sistema - Listado
Consulta Cliente
RUT.....: 2 - 7
NOMBRE.....: DOS
APE. PATERNO..: RAYA
APE. MATERNO..: SIETE
FEC NACIMIENTO: 01/01/1970
GENERO.....: M H=HOMBRE; M:MUJER
DIRECCION.....: DIRECCION DE DOS RAYA SIETE
CIUDAD.....: SANTIAGO
TELEFONOS.....: +5698761234
EMAIL.....: DOSRAYASIETE@GMAIL.COM
SALIR (S/N) : N
Símbolo del sistema - Listado
Listado Clientes
Posicionar Rut : 0000000000
LIN RUT NOMBRE
1 1-9 UNO RAYA NUEVE
2 2-7 DOS RAYA SIETE
SALIR (S/N) : n LINEA :02 Opcion C=Consultar; B=Borrar : b


Símbolo del sistema - Listado
Listado Clientes
Posicionar Rut : 0000000000
LIN RUT NOMBRE
1 1-9 UNO RAYA NUEVE
SALIR (S/N) : LINEA :00 Opcion C=Consultar; B=Borrar :

NO COPIA

Listado de Registros 2 (ListadoClientes2.CBL)

El siguiente código realiza lo mismo que el anterior, sin embargo lo que cambia es el Screen Section y la forma que se carga en el arreglo.

```
******************************************************************
* Author: Jorge Duarte
* Date: 02-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. ListadoClientes2.

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT datos-usuarios
         ASSIGN TO
     "C:\PgmCobols\Data\clientes.dat"
         ORGANIZATION IS INDEXED
         ACCESS MODE IS DYNAMIC
         RECORD KEY IS usuario-rut
         FILE STATUS IS FS-USUARIOS.

 DATA DIVISION.
 FILE SECTION.
 FD  datos-usuarios.
     copy "C:\PgmCobols\FD\FDCliente.cbl".

 WORKING-STORAGE SECTION.
 01  fin-de-archivo     PIC X VALUE 'N'.
 01  opcion             PIC X(1).
 01  continuar          PIC X VALUE 'Y'.
 01  usuario-clave      PIC X(10).
 01  done               PIC X VALUE 'N'.
 01  FS-USUARIOS        PIC XX.
 01  SALIR              PIC X.

 01 VARIABLES-DE-TRABAJO.
     05 BLANCO          PIC X.
     05 RUT-POS         PIC 9(10).
     05 WS-NOMBRE       PIC X(50).
     05 POS-ARR         PIC 9(3).
     05 WS-LINEA        PIC 9(3).
     05 WS-LIN          PIC 9(2).
     05 WS-OPC          PIC X.
     05 ARR-DETALLE OCCURS 15 TIMES.
        10 ARR-LIN      PIC 9(3).
        10 ARR-RUT      PIC 9(10).
        10 ARR-GUI      PIC X.
        10 ARR-DV       PIC X.
        10 ARR-NOM      PIC X(50).
        10 ARR-DET      PIC X(80).

 01 PAR-OUT-BORRAR.
     05 LK-BORRADO    PIC X.
*       S = Se Borro Cliente
*       N = No Borrado
```

NO COPIAR

```
    05 LK-STATUS     PIC X(02).

SCREEN SECTION.
01  PANTALLA-ENTRADA.
  03  BLANK SCREEN.
  03  LINE 1 COL 30  VALUE "Listado Clientes".
  03  LINE 3 COL 5   VALUE "Posicionar Rut : ".
  03  LINE 3 COL 24 PIC 9(10) USING RUT-POS.

01 PANTALLA-VISUALIZACION.
  03  LINE 5 COL 01 VALUE "====================================".
  03  LINE 5 COL 37 VALUE "====================================".
  03  LINE 5 COL 73 VALUE "=====".
  03  LINE 6 COL 01 VALUE "LIN RUT           NOMBRE".
  03  LINE 7 COL 01 VALUE "====================================".
  03  LINE 7 COL 37 VALUE "====================================".
  03  LINE 7 COL 73 VALUE "=====".

  03  LINE 08 COL 1  PIC X(80) FROM ARR-DET(01).
  03  LINE 09 COL 1  PIC X(80) FROM ARR-DET(02).
  03  LINE 10 COL 1  PIC X(80) FROM ARR-DET(03).
  03  LINE 11 COL 1  PIC X(80) FROM ARR-DET(04).
  03  LINE 12 COL 1  PIC X(80) FROM ARR-DET(05).
  03  LINE 13 COL 1  PIC X(80) FROM ARR-DET(06).
  03  LINE 14 COL 1  PIC X(80) FROM ARR-DET(07).
  03  LINE 15 COL 1  PIC X(80) FROM ARR-DET(08).
  03  LINE 16 COL 1  PIC X(80) FROM ARR-DET(09).
  03  LINE 17 COL 1  PIC X(80) FROM ARR-DET(10).
  03  LINE 18 COL 1  PIC X(80) FROM ARR-DET(11).
  03  LINE 19 COL 1  PIC X(80) FROM ARR-DET(12).
  03  LINE 20 COL 1  PIC X(80) FROM ARR-DET(13).
  03  LINE 21 COL 1  PIC X(80) FROM ARR-DET(14).
  03  LINE 22 COL 1  PIC X(80) FROM ARR-DET(15).


01 PANTALLA-SALIR.
  03 LINE 24 COL 5 VALUE "SALIR (S/N) : ".
  03 LINE 24 COL 19 PIC X USING SALIR.
  03 LINE 24 COL 21 VALUE "LINEA :".
  03 LINE 24 COL 28 PIC 9(02) USING WS-LIN.
  03 LINE 24 COL 31 VALUE "Opcion C=Consultar; B=Borrar :".
  03 LINE 24 COL 64 PIC X USING WS-OPC.

PROCEDURE DIVISION.
INICIO.
    OPEN INPUT datos-usuarios
    MOVE 0      TO RUT-POS
    MOVE "N"    TO SALIR
    PERFORM UNTIL SALIR = "S" OR "s"
       DISPLAY PANTALLA-ENTRADA
       ACCEPT  PANTALLA-ENTRADA

       PERFORM mostrar-todos-los-usuarios
       PERFORM PEDIR-SALIR
    END-PERFORM
    CLOSE datos-usuarios
    STOP RUN.

mostrar-todos-los-usuarios.
    PERFORM cargar-usuarios

    DISPLAY PANTALLA-ENTRADA
    ACCEPT  PANTALLA-VISUALIZACION
```

NO COPIAR

```
     .

cargar-usuarios.
    PERFORM START-READING

    PERFORM VARYING POS-ARR FROM 1 BY 1 UNTIL POS-ARR > 15
       MOVE ZEROES   TO ARR-LIN(POS-ARR)
                        ARR-RUT(POS-ARR)
       MOVE SPACES   TO ARR-DV (POS-ARR)
                        ARR-GUI(POS-ARR)
                        ARR-NOM(POS-ARR)
                        ARR-DET(POS-ARR)
    END-PERFORM

    MOVE 1  TO POS-ARR

    PERFORM UNTIL fin-de-archivo = 'Y' OR POS-ARR > 15
        READ datos-usuarios
            AT END
                MOVE 'Y' TO  fin-de-archivo
            NOT AT END

                MOVE POS-ARR         TO ARR-LIN    (POS-ARR)
                MOVE usuario-rut     TO ARR-RUT    (POS-ARR)
                MOVE "-"             TO ARR-GUI    (POS-ARR)
                MOVE usuario-dv      TO ARR-DV     (POS-ARR)
                STRING usuario-nombre DELIMITED BY " "
                     " "              DELIMITED BY SIZE
                     usuario-apepat DELIMITED BY " "
                     " "              DELIMITED BY SIZE
                     usuario-apemat DELIMITED BY " "
                                      INTO ARR-NOM    (POS-ARR)
                STRING POS-ARR            DELIMITED BY SIZE
                       " "                DELIMITED BY SIZE
                       usuario-rut        DELIMITED BY SIZE
                       "-"                DELIMITED BY SIZE
                       usuario-dv         DELIMITED BY SIZE
                       " "                DELIMITED BY SIZE
                       ARR-NOM(POS-ARR)   DELIMITED BY SIZE
                                      INTO ARR-DET(POS-ARR)

                ADD  1               TO POS-ARR
        END-READ
    END-PERFORM

    .
START-READING.
    MOVE 'N'        TO fin-de-archivo
    MOVE RUT-POS    TO usuario-rut
    START datos-usuarios KEY IS NOT < usuario-rut
        INVALID KEY
            DISPLAY "Error al iniciar lectura de registros."
            MOVE "Y"    TO fin-de-archivo
    END-START.

PEDIR-SALIR.
    MOVE " "            TO SALIR
    PERFORM UNTIL SALIR = "S" OR "s" or "N" or "n"
       DISPLAY PANTALLA-SALIR
       ACCEPT  PANTALLA-SALIR
       IF WS-LIN NOT = 0 AND WS-LIN < 16 THEN
```

NO COPIAR

```
            IF WS-OPC = "C" OR "c" THEN
               PERFORM LLAMA-CONSULTA
               MOVE ZEROES  TO WS-LIN
               MOVE SPACES  TO WS-OPC
               MOVE "N" TO SALIR
            END-IF

            IF WS-OPC = "B" OR "b" THEN
               PERFORM ELIMINA-CLIENTE
               MOVE ZEROES  TO WS-LIN
               MOVE SPACES  TO WS-OPC
               MOVE "N" TO SALIR
            END-IF
         END-IF
      END-PERFORM
      .

LLAMA-CONSULTA.
      IF ARR-RUT(WS-LIN) NOT = 0 THEN
          CALL "ConsultaCliente" USING ARR-RUT(WS-LIN)
      END-IF
      .

 ELIMINA-CLIENTE.
      IF ARR-RUT(WS-LIN) NOT = 0 THEN
          CALL "BorrarCliente" USING ARR-RUT(WS-LIN)
                                     PAR-OUT-BORRAR
      END-IF
      .
END PROGRAM ListadoClientes2.
```

NO COPIAR

Generar Archivo para Calendario (GenCalendario.CBL)

Este programa parte desde el 01-01-1925, y genera registros hasta el 31-12-2100, esto puede ser modificado de acuerdo a las necesidades, como consideración se consideran los Domingos como Feriados, se puede generar un mantenedor para ir agregando los feriados necesarios para cada país, o también agregar si es día Hábil, que puede ser utilizado para generar una Rutina para sumar Días Hábiles.

Como consideración se debe tener creado el Archivo **calendario.dat**.

```
******************************************************************
* Author: JORGE DUARTE URZUA
* Date: 17-07-2025
* Purpose: ENSEÑANZA, SE DEJA TODOS LOS DOMINGOS COMO FERIADO
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. GenCalendario.
 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT calendario
         ASSIGN TO
     "C:\PgmCobols\Data\calendario.dat"
         ORGANIZATION IS INDEXED
         ACCESS MODE IS DYNAMIC
         RECORD KEY IS calendario-fecha
         FILE STATUS IS FS-CALENDARIO.
 DATA DIVISION.
 FILE SECTION.
 FD  calendario.
     copy "C:\PgmCobols\FD\FDCalendario.cpy".

 WORKING-STORAGE SECTION.
 01  FS-CALENDARIO          PIC XX.
 01  FECHA-INI              PIC 9(08) VALUE 19250101.
 01  FECHA-TOPE             PIC 9(08) VALUE 21010101.
 01  DIA-INI                PIC 9     VALUE 5.

 01 PARAMETROS-ENTRADA.
     05 FECHA-YYYYMMDD PIC X(08).
     05 SUM-RES-DIAS   PIC S9(03).
 01 PARAMETRO-SALIDA.
     05 FECHA-VALIDA          PIC X(01).
     05 NUEVA-FECHA-YYYYMMDD PIC X(08).

 PROCEDURE DIVISION.
 MAIN-PROCEDURE.
      DISPLAY "FECHA-INI  : " FECHA-INI
      OPEN OUTPUT calendario

      PERFORM UNTIL FECHA-INI = FECHA-TOPE  OR
                    FS-CALENDARIO NOT = "00"
          MOVE FECHA-INI   TO calendario-fecha
          MOVE DIA-INI     TO calendario-dia-sem
          IF DIA-INI = 1 THEN
             MOVE "S"      TO calendario-feriado
```

NO COPIAR

```
           ELSE
              MOVE "N"      TO calendario-feriado
           END-IF
           WRITE r-calendario END-WRITE
           IF FS-CALENDARIO = "00"
              MOVE FECHA-INI        TO FECHA-YYYYMMDD
              MOVE 1                TO SUM-RES-DIAS
              CALL "SumarDiasFecha" USING PARAMETROS-ENTRADA
                                          PARAMETRO-SALIDA
              MOVE NUEVA-FECHA-YYYYMMDD TO FECHA-INI
              ADD 1            TO DIA-INI
              IF DIA-INI > 7 THEN
                 MOVE 1        TO DIA-INI
              END-IF
           END-IF
           DISPLAY "FECHA-INI (GEN) : " FECHA-INI
        END-PERFORM
        DISPLAY "ULTIMA FECHA " FECHA-INI

        CLOSE calendario
        STOP RUN.
END PROGRAM GenCalendario.
```

NO COPIAR

Mostrar Fecha Calendario (FechasCalendario.CBL)

```
******************************************************************
* Author: Jorge Duarte
* Date: 17-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. FechasCalendario.

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT calendario
         ASSIGN TO
     "C:\PgmCobols\Data\calendario.dat"
         ORGANIZATION IS INDEXED
         ACCESS MODE IS DYNAMIC
         RECORD KEY IS calendario-fecha
         FILE STATUS IS FS-CALENDARIO.

 DATA DIVISION.
 FILE SECTION.
 FD  calendario.
     copy "C:\PgmCobols\FD\FDCalendario.cpy".

 WORKING-STORAGE SECTION.
 01  fin-de-archivo      PIC X VALUE 'N'.
 01  opcion              PIC X(1).
 01  continuar           PIC X VALUE 'Y'.
 01  usuario-clave       PIC X(10).
 01  done                PIC X VALUE 'N'.
 01  FS-CALENDARIO       PIC XX.
 01  SALIR               PIC X.

 01 VARIABLES-DE-TRABAJO.
     05 BLANCO           PIC X.
     05 WS-FECHA         PIC X(08).
     05 WS-DIA           PIC X(10).
     05 FECHA-POS        PIC 9(08).
     05 R-FECHA-POS REDEFINES FECHA-POS.
        10 F-POS-DIA     PIC 9(02).
        10 F-POS-MES     PIC 9(02).
        10 F-POS-ANO     PIC 9(04).
     05 FECHA-AMD.
        10 F-POS-ANO     PIC 9(04).
        10 F-POS-MES     PIC 9(02).
        10 F-POS-DIA     PIC 9(02).
     05 WS-NOMBRE        PIC X(50).
     05 POS-ARR          PIC 9(3).
     05 WS-LINEA         PIC 9(3).
     05 WS-LIN           PIC 9(2).
     05 WS-OPC           PIC X.
     05 ARR-DETALLE OCCURS 15 TIMES.
        10 ARR-LIN       PIC 9(3).
        10 ARR-FEC       PIC X(10).
        10 ARR-DIASEM    PIC X(10).
        10 ARR-FERIADO   PIC X.
        10 ARR-DET       PIC X(50).

 01 PAR-OUT-BORRAR.
     05 LK-BORRADO    PIC X.
```

NO COPIAR

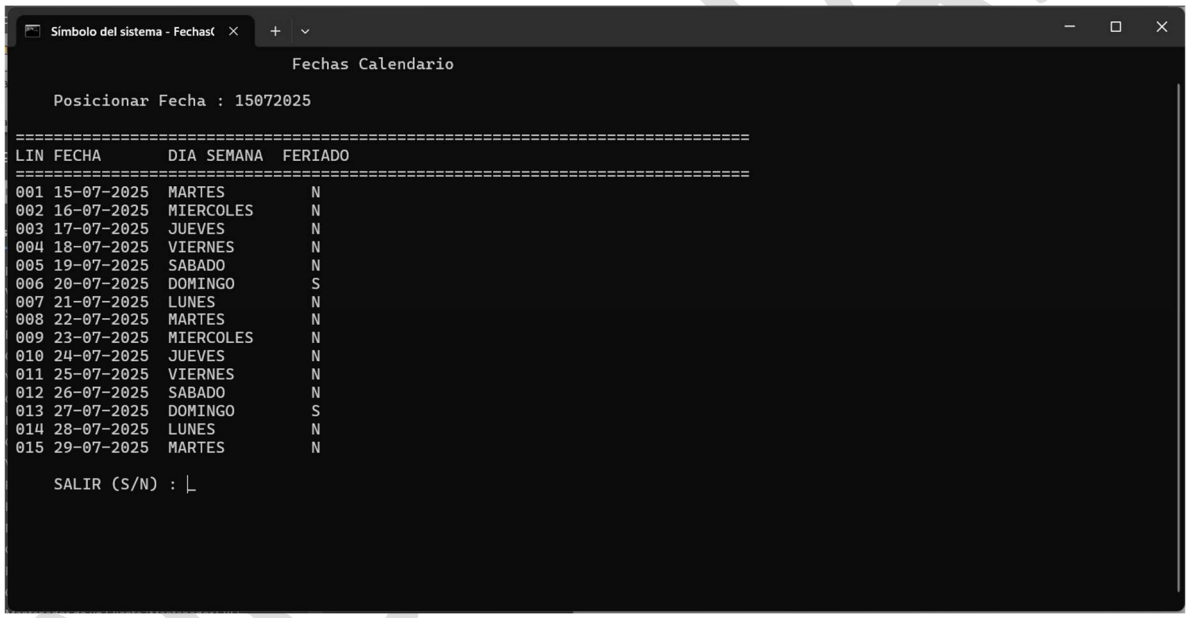
```
*       S = Se Borro Cliente
*       N = No Borrado
     05 LK-STATUS     PIC X(02).

 SCREEN SECTION.
 01  PANTALLA-ENTRADA.
   03  BLANK SCREEN.
   03  LINE 1 COL 30  VALUE "Fechas Calendario".
   03  LINE 3 COL 5   VALUE "Posicionar Fecha : ".
   03  LINE 3 COL 24 PIC 9(08) USING FECHA-POS.

 01 PANTALLA-VISUALIZACION.
   03  LINE 5 COL 01 VALUE "====================================".
   03  LINE 5 COL 37 VALUE "====================================".
   03  LINE 5 COL 73 VALUE "=====".
   03  LINE 6 COL 01 VALUE "LIN FECHA       DIA SEMANA  FERIADO".
   03  LINE 7 COL 01 VALUE "====================================".
   03  LINE 7 COL 37 VALUE "====================================".
   03  LINE 7 COL 73 VALUE "=====".

   03  LINE 08 COL 1  PIC X(80) FROM ARR-DET(01).
   03  LINE 09 COL 1  PIC X(80) FROM ARR-DET(02).
   03  LINE 10 COL 1  PIC X(80) FROM ARR-DET(03).
   03  LINE 11 COL 1  PIC X(80) FROM ARR-DET(04).
   03  LINE 12 COL 1  PIC X(80) FROM ARR-DET(05).
   03  LINE 13 COL 1  PIC X(80) FROM ARR-DET(06).
   03  LINE 14 COL 1  PIC X(80) FROM ARR-DET(07).
   03  LINE 15 COL 1  PIC X(80) FROM ARR-DET(08).
   03  LINE 16 COL 1  PIC X(80) FROM ARR-DET(09).
   03  LINE 17 COL 1  PIC X(80) FROM ARR-DET(10).
   03  LINE 18 COL 1  PIC X(80) FROM ARR-DET(11).
   03  LINE 19 COL 1  PIC X(80) FROM ARR-DET(12).
   03  LINE 20 COL 1  PIC X(80) FROM ARR-DET(13).
   03  LINE 21 COL 1  PIC X(80) FROM ARR-DET(14).
   03  LINE 22 COL 1  PIC X(80) FROM ARR-DET(15).

 01 PANTALLA-SALIR.
   03 LINE 24 COL 5 VALUE "SALIR (S/N) : ".
   03 LINE 24 COL 19 PIC X USING SALIR.


 PROCEDURE DIVISION.
 INICIO.
     OPEN INPUT calendario
     MOVE 0      TO FECHA-POS
     MOVE "N"    TO SALIR
     PERFORM UNTIL SALIR = "S" OR "s"
         DISPLAY PANTALLA-ENTRADA
         ACCEPT  PANTALLA-ENTRADA

         PERFORM mostrar-fechas
         PERFORM PEDIR-SALIR

     END-PERFORM
     CLOSE calendario
     STOP RUN.

 mostrar-fechas.

     PERFORM cargar-fechas

     DISPLAY PANTALLA-ENTRADA
     ACCEPT  PANTALLA-VISUALIZACION
```

NO COPIAR

```
     .

cargar-fechas.
    PERFORM START-READING

    PERFORM VARYING POS-ARR FROM 1 BY 1 UNTIL POS-ARR > 15
       MOVE ZEROES   TO ARR-LIN    (POS-ARR)
       MOVE SPACES   TO ARR-FEC    (POS-ARR)
                        ARR-DIASEM (POS-ARR)
                        ARR-FERIADO(POS-ARR)
                        ARR-DET    (POS-ARR)
    END-PERFORM

    MOVE 1  TO POS-ARR

    PERFORM UNTIL fin-de-archivo = 'Y' OR POS-ARR > 15 OR
            FS-CALENDARIO NOT = "00"

        READ calendario
            AT END
                MOVE 'Y' TO  fin-de-archivo
            NOT AT END

                MOVE POS-ARR           TO ARR-LIN    (POS-ARR)
                MOVE calendario-fecha  TO ARR-FEC    (POS-ARR)
                                          WS-FECHA

                EVALUATE calendario-dia-sem
                  WHEN 1 MOVE "DOMINGO"    TO WS-DIA
                  WHEN 2 MOVE "LUNES"      TO WS-DIA
                  WHEN 3 MOVE "MARTES"     TO WS-DIA
                  WHEN 4 MOVE "MIERCOLES"  TO WS-DIA
                  WHEN 5 MOVE "JUEVES"     TO WS-DIA
                  WHEN 6 MOVE "VIERNES"    TO WS-DIA
                  WHEN 7 MOVE "SABADO"     TO WS-DIA
                END-EVALUATE

                STRING  POS-ARR    DELIMITED SIZE
                        " "        DELIMITED SIZE
                        WS-FECHA(7:2) DELIMITED SIZE
                        "-"        DELIMITED SIZE
                        WS-FECHA(5:2) DELIMITED SIZE
                        "-"        DELIMITED SIZE
                        WS-FECHA(1:4) DELIMITED SIZE
                        "  "       DELIMITED SIZE
                        WS-DIA     DELIMITED SIZE
                        "     "    DELIMITED SIZE
                   calendario-feriado DELIMITED SIZE
                                   INTO ARR-DET    (POS-ARR)


                ADD  1              TO POS-ARR
        END-READ

    END-PERFORM

    .

START-READING.
    MOVE 'N'            TO fin-de-archivo

    MOVE CORR R-FECHA-POS TO FECHA-AMD
```

NO COPIAR

```
    MOVE FECHA-AMD        TO calendario-fecha

    START calendario KEY IS NOT < calendario-fecha
        INVALID KEY
            DISPLAY "Error al iniciar lectura de registros."
            MOVE "Y"    TO fin-de-archivo
    END-START.

PEDIR-SALIR.
    MOVE " "            TO SALIR
    PERFORM UNTIL SALIR = "S" OR "s" or "N" or "n"
       DISPLAY PANTALLA-SALIR
       ACCEPT  PANTALLA-SALIR

    END-PERFORM
    .

END PROGRAM FechasCalendario.
```


Símbolo del sistema - Fechas(
Fechas Calendario
Posicionar Fecha : 15072025
LIN FECHA DIA SEMANA FERIADO
001 15-07-2025 MARTES N
002 16-07-2025 MIERCOLES N
003 17-07-2025 JUEVES N
004 18-07-2025 VIERNES N
005 19-07-2025 SABADO N
006 20-07-2025 DOMINGO S
007 21-07-2025 LUNES N
008 22-07-2025 MARTES N
009 23-07-2025 MIERCOLES N
010 24-07-2025 JUEVES N
011 25-07-2025 VIERNES N
012 26-07-2025 SABADO N
013 27-07-2025 DOMINGO S
014 28-07-2025 LUNES N
015 29-07-2025 MARTES N
SALIR (S/N) :

Mostrar Calendario (MostrarCalendario.CBL)

```
******************************************************************
* Author: Jorge Duarte
* Date: 17-07-2025
* Purpose: Estudio
* Tectonics: cobc
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. MostrarCalendario.

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT calendario
         ASSIGN TO
     "C:\PgmCobols\Data\calendario.dat"
         ORGANIZATION IS INDEXED
         ACCESS MODE IS DYNAMIC
         RECORD KEY IS calendario-fecha
         FILE STATUS IS FS-CALENDARIO.

 DATA DIVISION.
 FILE SECTION.
 FD  calendario.
     copy "C:\PgmCobols\FD\FDCalendario.cpy".

 WORKING-STORAGE SECTION.
 01  fin-de-archivo      PIC X VALUE 'N'.
 01  opcion              PIC X(1).
 01  continuar           PIC X VALUE 'Y'.
 01  usuario-clave       PIC X(10).
 01  done                PIC X VALUE 'N'.
 01  FS-CALENDARIO       PIC XX.
 01  SALIR               PIC X.

 01 VARIABLES-DE-TRABAJO.
     05 BLANCO           PIC X.
     05 PEDIR-MES        PIC 9(02).
     05 PEDIR-ANO        PIC 9(04).
     05 WS-FECHA         PIC X(08).
     05 WS-DIA           PIC X(10).
     05 FECHA-POS        PIC 9(08).
     05 R-FECHA-POS REDEFINES FECHA-POS.
        10 F-POS-DIA     PIC 9(02).
        10 F-POS-MES     PIC 9(02).
        10 F-POS-ANO     PIC 9(04).
     05 FECHA-AMD.
        10 F-POS-ANO     PIC 9(04).
        10 F-POS-MES     PIC 9(02).
        10 F-POS-DIA     PIC 9(02).
     05 WS-NOMBRE        PIC X(50).
     05 POS-ARR          PIC 9(3).
     05 WS-LINEA         PIC 9(3).
     05 WS-LIN           PIC 9(2).
     05 WS-OPC           PIC X.
     05 POS-I-DIAS       PIC 9.
     05 ARR-DIA OCCURS 07 TIMES.
        10 ARR-DIA-S    PIC ZZ.
     05 ARR-MES OCCURS 05 TIMES.
        10 ARR-DET    PIC X(38).

     05 WS-PEDIR-X      PIC X(06).
```

NO COPIAR

```
    05 WS-PEDIR REDEFINES WS-PEDIR-X PIC 9(06).

    05 WS-FEC-ARC.
       10 WS-FEC-YYYYMM PIC 9(06).
       10 WS-FEC-DD     PIC 9(02).



SCREEN SECTION.
01  PANTALLA-ENTRADA.
  03  BLANK SCREEN.
  03  LINE 1 COL 40  VALUE "Calendario".
  03  LINE 3 COL 5   VALUE "Mes : ".
  03  LINE 3 COL 10 PIC 9(02) USING PEDIR-MES.
  03  LINE 3 COL 15   VALUE "Año : ".
  03  LINE 3 COL 24 PIC 9(04) USING PEDIR-ANO.

01 PANTALLA-VISUALIZACION.
  03  LINE 5 COL 10 VALUE "+====+====+====+====+====+====+----+".
  03  LINE 6 COL 10 VALUE "| LN | MA | MI | JU | VI | SA | DO |".
  03  LINE 7 COL 10 VALUE "+====+====+====+====+====+====+====+".

  03  LINE 08 COL 10  PIC X(80) FROM ARR-DET(01).
  03  LINE 09 COL 10  PIC X(80) FROM ARR-DET(02).
  03  LINE 10 COL 10  PIC X(80) FROM ARR-DET(03).
  03  LINE 11 COL 10  PIC X(80) FROM ARR-DET(04).
  03  LINE 12 COL 10  PIC X(80) FROM ARR-DET(05).
  03 LINE 13 COL 10 VALUE "+====+====+====+====+====+====+====+".

01 PANTALLA-SALIR.
  03 LINE 24 COL 5 VALUE "SALIR (S/N) : ".
  03 LINE 24 COL 19 PIC X USING SALIR.


PROCEDURE DIVISION.
INICIO.
    OPEN INPUT calendario
    MOVE 0      TO FECHA-POS
    MOVE "N"    TO SALIR
    PERFORM UNTIL SALIR = "S" OR "s"
        DISPLAY PANTALLA-ENTRADA
        ACCEPT  PANTALLA-ENTRADA

        PERFORM mostrar-fechas
        PERFORM PEDIR-SALIR

    END-PERFORM
    CLOSE calendario
    STOP RUN.

mostrar-fechas.

    PERFORM cargar-fechas

    DISPLAY PANTALLA-ENTRADA
    ACCEPT  PANTALLA-VISUALIZACION

     .

cargar-fechas.
    PERFORM START-READING
```

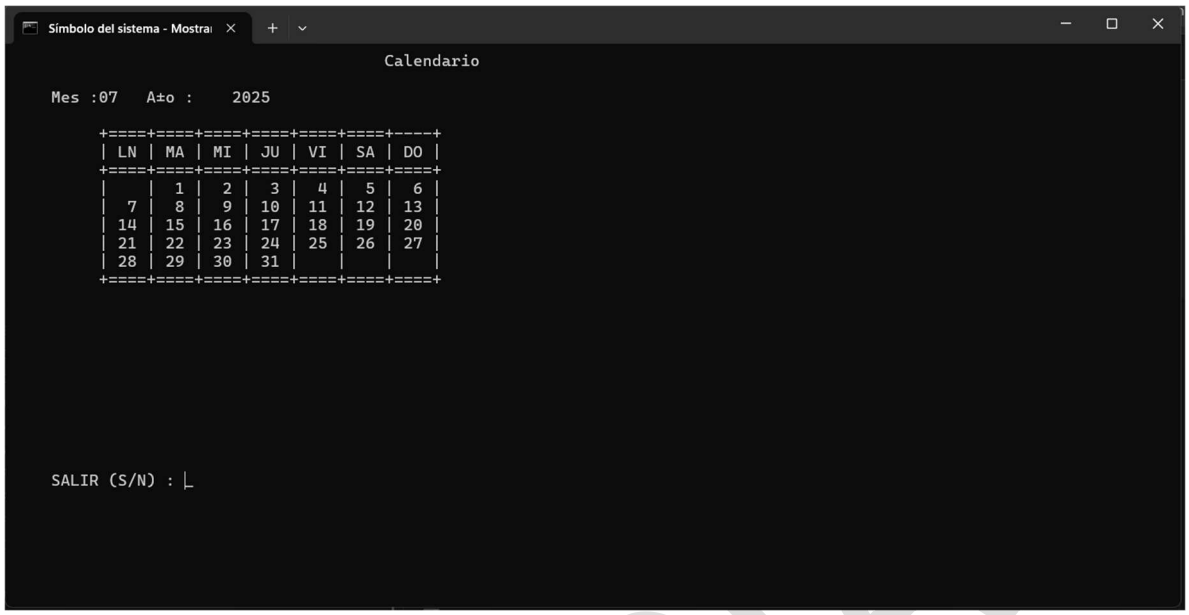
NO COPIAR

```
PERFORM VARYING POS-ARR FROM 1 BY 1 UNTIL POS-ARR > 5
   MOVE SPACES   TO ARR-DET    (POS-ARR)
END-PERFORM

MOVE ZEROES   TO ARR-DIA-S(1)
                 ARR-DIA-S(2)
                 ARR-DIA-S(3)
                 ARR-DIA-S(4)
                 ARR-DIA-S(5)
                 ARR-DIA-S(6)
                 ARR-DIA-S(7)

MOVE 1  TO POS-ARR

PERFORM UNTIL fin-de-archivo = 'Y' OR POS-ARR > 5 OR
        FS-CALENDARIO NOT = "00"

    READ calendario
        AT END
            MOVE 'Y' TO  fin-de-archivo
        NOT AT END
        MOVE calendario-fecha TO WS-FEC-ARC

        IF WS-FEC-YYYYMM NOT = WS-PEDIR THEN
           MOVE "Y" TO fin-de-archivo
        ELSE
            EVALUATE calendario-dia-sem
              WHEN 1 MOVE 7      TO POS-I-DIAS
              WHEN 2 MOVE 1      TO POS-I-DIAS
              WHEN 3 MOVE 2      TO POS-I-DIAS
              WHEN 4 MOVE 3      TO POS-I-DIAS
              WHEN 5 MOVE 4      TO POS-I-DIAS
              WHEN 6 MOVE 5      TO POS-I-DIAS
              WHEN 7 MOVE 6      TO POS-I-DIAS
            END-EVALUATE

            MOVE WS-FEC-DD TO ARR-DIA-S(POS-I-DIAS)
            IF POS-I-DIAS = 7 THEN

              STRING  "| "           DELIMITED SIZE
                      ARR-DIA-S(1)    DELIMITED SIZE
                      " | "           DELIMITED SIZE
                      ARR-DIA-S(2)    DELIMITED SIZE
                      " | "           DELIMITED SIZE
                      ARR-DIA-S(3)    DELIMITED SIZE
                      " | "           DELIMITED SIZE
                      ARR-DIA-S(4)    DELIMITED SIZE
                      " | "           DELIMITED SIZE
                      ARR-DIA-S(5)    DELIMITED SIZE
                      " | "           DELIMITED SIZE
                      ARR-DIA-S(6)    DELIMITED SIZE
                      " | "           DELIMITED SIZE
                      ARR-DIA-S(7)    DELIMITED SIZE
                      " |"            DELIMITED SIZE
                                 INTO ARR-DET    (POS-ARR)

              MOVE ZEROES   TO ARR-DIA-S(1)
                               ARR-DIA-S(2)
                               ARR-DIA-S(3)
                               ARR-DIA-S(4)
                               ARR-DIA-S(5)
                               ARR-DIA-S(6)
                               ARR-DIA-S(7)
```

NO COPIAR

```
                    ADD  1               TO POS-ARR
                  END-IF
               END-IF
           END-READ

       END-PERFORM

*      AGREGAMOS LO ULTIMO
       IF POS-ARR < 6 THEN
          STRING  "| "                          DELIMITED SIZE
                           ARR-DIA-S(1)    DELIMITED SIZE
                           " | "           DELIMITED SIZE
                           ARR-DIA-S(2)    DELIMITED SIZE
                           " | "           DELIMITED SIZE
                           ARR-DIA-S(3)    DELIMITED SIZE
                           " | "           DELIMITED SIZE
                           ARR-DIA-S(4)    DELIMITED SIZE
                           " | "           DELIMITED SIZE
                           ARR-DIA-S(5)    DELIMITED SIZE
                           " | "           DELIMITED SIZE
                           ARR-DIA-S(6)    DELIMITED SIZE
                           " | "           DELIMITED SIZE
                           ARR-DIA-S(7)    DELIMITED SIZE
                           " |"            DELIMITED SIZE
                                    INTO ARR-DET    (POS-ARR)
       END-IF
       .

   START-READING.
       MOVE 'N'              TO fin-de-archivo

       MOVE PEDIR-MES        TO F-POS-MES OF R-FECHA-POS
                                WS-PEDIR-X(5:2)
       MOVE PEDIR-ANO        TO F-POS-ANO OF R-FECHA-POS
                                WS-PEDIR-X(1:4)
       MOVE 01               TO F-POS-DIA OF R-FECHA-POS



       MOVE CORR R-FECHA-POS TO FECHA-AMD

       MOVE FECHA-AMD         TO calendario-fecha

       START calendario KEY IS NOT < calendario-fecha
         INVALID KEY
             DISPLAY "Error al iniciar lectura de registros."
             MOVE "Y"    TO fin-de-archivo

       END-START.

   PEDIR-SALIR.
       MOVE " "            TO SALIR
       PERFORM UNTIL SALIR = "S" OR "s" or "N" or "n"
          DISPLAY PANTALLA-SALIR
          ACCEPT  PANTALLA-SALIR

       END-PERFORM
       .

   END PROGRAM MostrarCalendario.
```

NO COPIAR


Símbolo del sistema - Mostra
Calendario
Mes :07 Año : 2025
LN	MA	MI	JU	VI	SA	DO
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
SALIR (S/N) :

NO COPIA

Bonus Track AS400

En el siguiente Capitulo, se mostrarán desarrollo en ISERIES (AS400), donde se pueden ver Pantallas, Programas de Lenguaje de Control (CLP), Creación de Archivos, y Programas Cobol y Cobol SQL.

El desarrollo en AS400 es diferente al que hemos visto, ya que permite trabajar con Teclas de Funciones, SQL Embebido, y Pantallas complejas, incluidos Botones, que hacen más amigables los desarrollos.

Realizaremos algo similar a lo revisado en el capitulo anterior, y la estructura la definiremos de la siguiente forma:

- Crearemos una Biblioteca por ejemplo LIBFTE. (CRTLIB LIB(LIBFTE))
- Crearemos los Archivos para guardar los fuentes (CRTSRCPF)
    - QCBLSRC – Para guardar los Fuentes Cobol
    - QCLSRC – Para guardar los Fuentes CLP
    - QDDSSRC – Para guardar los fuentes asociados a los Archivos
    - QDSPSRC – Para Guardar los fuentes asociados a Pantallas

QCBLSRC	BORRARCLI	CBL	Borrar Clientes
	CONCLI	CBL	Consulta Cliente
	GRABARCLI	CBL	Grabar Clientes
	LARGOSTR	CBL	Calcula Largo de un String
	LISCLI	CBL	Listado de Clientes
	MANCLICBL	CBL	Mantenedor de Clientes
	VALFEC	CBL	Valida Fecha
	VALRUT	CBL	Valida Rut
	VERMSJ	CBL	Ver Mensaje
QDDSSRC	CLIENTES	PF	Tabla de Clientes
QDSPSRC	CONS01	DSPF	Consulta de Clientes
	LIST01	DSPF	Listado de Clientes
	MANT01	DSPF	Mantenedor de Clientes
	MENU01	DSPF	MENU
	VENMSG	DSPF	Ventana de Mensajes
QCLPSRC	MENUCLP	CLP	MENU DEL SISTEMA

NO COPIAR

Creación Archivo de Clientes (CLIENTES-PF)

```
A
A                                       UNIQUE
A          R CLIREG
A            RUT           10S 0       COLHDG('RUT DEL CLIENTE')
A            DV             1A         COLHDG('DV RUT')
A            NOMBRE        30A         COLHDG('NOMBRE CLIENTE')
A            APEPAT        30A         COLHDG('APE PATERNO')
A            APEMAT        30A         COLHDG('APE MATERNO')
A            FECNAC         8S 0       COLHDG('FECHA NACIMIENTO')
A            GENERO         1A         COLHDG('H=HOMBRE;M=MUJER')
A            DIRECC        50A         COLHDG('DIRECCION')
A            CIUDAD        50A         COLHDG('CIUDAD')
A            TELEFONO1     15A         COLHDG('TELEFONO 1')
A            TELEFONO2     15A         COLHDG('TELEFONO 2')
A            EMAIL        100A         COLHDG('EMAIL CLIENTE')
A            FECCRE         8S         COLHDG('FECHA CREACION')
A          K RUT
```

NO COPIAR

Creación Pantalla Consulta de Clientes (CONS01)

```
A                                      DSPSIZ(24 80 *DS3)
A                                      INDARA
A                                      CF03(03 'Salir')
A  50                                  CF07(07 'Grabar')
A          R R01
A*%%TS  SD  20250714  165849  IG0JDUEX    REL-V7R2M0  5770-WDS
A                                   1 59DATE
A                                      EDTCDE(Y)
A                                   2 59TIME
A                                   3 57USER
A                                   6 10'Rut Cliente  :'
A                                      DSPATR(HI)
A            RUT       R        B  6 26REFFLD(CLIREG/RUT LIBJDU/CLIENTES)
A                                      EDTCDE(Z)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A                                   6 37'-'
A            DV        R        B  6 39REFFLD(CLIREG/DV LIBJDU/CLIENTES)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A                                   7  8'Nombre Cliente :'
A                                      DSPATR(HI)
A            NOMBRE    R        B  7 25REFFLD(CLIREG/NOMBRE LIBJDU/CLIENTE-
A                                      S)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A                                   8  7'Apellido Paterno:'
A                                      DSPATR(HI)
A                                   9  7'Apellido Materno:'
A                                      DSPATR(HI)
A            APEPAT    R        B  8 25REFFLD(CLIREG/APEPAT LIBJDU/CLIENTE-
A                                      S)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A            APEMAT    R        B  9 25REFFLD(CLIREG/APEMAT LIBJDU/CLIENTE-
A                                      S)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A                                  10  7'Fecha Nacimiento:'
A                                      DSPATR(HI)
A            FECNAC    R        B 10 25REFFLD(CLIREG/FECNAC LIBJDU/CLIENTE-
A                                      S)
A                                      EDTCDE(Y)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A                                  11 16'Género :'
A                                      DSPATR(HI)
A            GENERO    R        B 11 25REFFLD(CLIREG/GENERO LIBJDU/CLIENTE-
A                                      S)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A                                  11 28'(H=Hombre; M=Mujer)'
A                                  12 13'Dirección :'
A                                      DSPATR(HI)
A            DIRECC    R        B 12 25REFFLD(CLIREG/DIRECC LIBJDU/CLIENTE-
A                                      S)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A                                  13 13'Ciudad    :'
A                                      DSPATR(HI)
A            CIUDAD    R        B 13 25REFFLD(CLIREG/CIUDAD LIBJDU/CLIENTE-
```

NO COPIAR

```
A                                       S)
A  99                                   DSPATR(UL)
A                                       DSPATR(PR)
A                                    14 13'Teléfonos :'
A                                       DSPATR(HI)
A            TELEFONO1 R        B 14 25REFFLD(CLIREG/TELEFONO1 LIBJDU/CLIE-
A                                       NTES)
A  99                                   DSPATR(UL)
A                                       DSPATR(PR)
A            TELEFONO2 R        B 14 42REFFLD(CLIREG/TELEFONO2 LIBJDU/CLIE-
A                                       NTES)
A  99                                   DSPATR(UL)
A                                       DSPATR(PR)
A                                    15 13'Email     :'
A                                       DSPATR(HI)
A            EMAIL     R        B 15 25REFFLD(CLIREG/EMAIL LIBJDU/CLIENTES)
A  99                                   DSPATR(UL)
A                                       DSPATR(PR)
A            BOTONES        2Y 0B 22  5PSHBTNFLD
A                                       PSHBTNCHC(1 'F3-Salir' CF03)
A  50                                   PSHBTNCHC(2 'F7-Grabar' CF07)
A                                     2 31'Consulta de Clientes'
A                                       DSPATR(HI)
A                                       DSPATR(RI)
```

NO COPIAR

Creación Pantalla Listado de Clientes (LIST01)

```
A                                      DSPSIZ(24 80 *DS3)
A                                      INDARA
A                                      CF03(03 'Salir')
A                                      CF06(06 'Crear Nuevo Cliente')
A          R S01                       SFL
A*%%TS  SD  20250714  170856  IG0JDUEX    REL-V7R2M0  5770-WDS
A                                      SFLNXTCHG
A            OPCS01         1   B  9  3
A            RUTS01        10Y 0B  9  6
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A                                      EDTCDE(Z)
A                                   9 17'-'
A            DVS01          1   B  9 19
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A            NOMS01        50   B  9 22
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A          R C01                       SFLCTL(S01)
A                                      OVERLAY
A  30                                  SFLDSP
A  31                                  SFLDSPCTL
A  32                                  SFLCLR
A  30                                  SFLEND(*SCRBAR *SCRBAR)
A                                      SFLSIZ(0100)
A                                      SFLPAG(0012)
A                                   1 65DATE
A                                      EDTCDE(Y)
A                                   2 65TIME
A                                   3 63USER
A                                   2 31'Listado de Clientes'
A                                      DSPATR(RI)
A                                      DSPATR(HI)
A                                   5  4'Posicionar Rut : '
A                                      DSPATR(HI)
A            RUTPOS        10Y 0B  5 23EDTCDE(Z)
A                                   8  2'Opc Rut             Nombre        -
A                                                                          -
A                                        '
A                                      DSPATR(UL)
A                                      DSPATR(HI)
A                                   5 39'Opción : C=Consultar B=Borrar'
A                                      DSPATR(HI)
A          R R01
A                                      OVERLAY
A                                  22  2'                                  -
A                                                                          -
A                                                    '
A                                      DSPATR(UL)
A                                  23  2'F3-Salir'
A                                  23 18'F6-Crear Cliente'
```

NO COPIAR

Creación Pantalla Mantenedor de Clientes (MANT01)

```
A                                      DSPSIZ(24 80 *DS3)
A                                      INDARA
A                                      CF03(03 'Salir')
A  50                                  CF07(07 'Grabar')
A          R R01
A                                    1 59DATE
A                                      EDTCDE(Y)
A                                    2 59TIME
A                                    3 57USER
A                                    2 31'Ingreso de Clientes'
A                                      DSPATR(HI)
A                                    6 10'Rut Cliente  :'
A                                      DSPATR(HI)
A            RUT       R        B  6 26REFFLD(CLIREG/RUT LIBJDU/CLIENTES)
A                                      EDTCDE(Z)
A                                    6 37'-'
A            DV        R        B  6 39REFFLD(CLIREG/DV LIBJDU/CLIENTES)
A                                    7  8'Nombre Cliente :'
A                                      DSPATR(HI)
A            NOMBRE    R        B  7 25REFFLD(CLIREG/NOMBRE LIBJDU/CLIENTE-
A                                      S)
A                                    8  7'Apellido Paterno:'
A                                      DSPATR(HI)
A                                    9  7'Apellido Materno:'
A                                      DSPATR(HI)
A            APEPAT    R        B  8 25REFFLD(CLIREG/APEPAT LIBJDU/CLIENTE-
A                                      S)
A            APEMAT    R        B  9 25REFFLD(CLIREG/APEMAT LIBJDU/CLIENTE-
A                                      S)
A                                   10  7'Fecha Nacimiento:'
A                                      DSPATR(HI)
A            FECNAC    R        B 10 25REFFLD(CLIREG/FECNAC LIBJDU/CLIENTE-
A                                      S)
A                                      EDTCDE(Y)
A                                   11 16'Género :'
A                                      DSPATR(HI)
A            GENERO    R        B 11 25REFFLD(CLIREG/GENERO LIBJDU/CLIENTE-
A                                      S)
A                                   11 28'(H=Hombre; M=Mujer)'
A                                   12 13'Dirección :'
A                                      DSPATR(HI)
A            DIRECC    R        B 12 25REFFLD(CLIREG/DIRECC LIBJDU/CLIENTE-
A                                      S)
A                                   13 13'Ciudad    :'
A                                      DSPATR(HI)
A            CIUDAD    R        B 13 25REFFLD(CLIREG/CIUDAD LIBJDU/CLIENTE-
A                                      S)
A                                   14 13'Teléfonos :'
A                                      DSPATR(HI)
A            TELEFONO1 R        B 14 25REFFLD(CLIREG/TELEFONO1 LIBJDU/CLIE-
A                                      NTES)
A            TELEFONO2 R        B 14 42REFFLD(CLIREG/TELEFONO2 LIBJDU/CLIE-
A                                      NTES)
A                                   15 13'Email     :'
A                                      DSPATR(HI)
A            EMAIL     R        B 15 25REFFLD(CLIREG/EMAIL LIBJDU/CLIENTES)
A            BOTONES        2Y 0B 22  5PSHBTNFLD
A                                      PSHBTNCHC(1 'F3-Salir' CF03)
A  50                                  PSHBTNCHC(2 'F7-Grabar' CF07)
```

NO COPIAR

Creación Pantalla Menú (MENU01)

```
A                                      DSPSIZ(24 80 *DS3)
A                                      INDARA
A                                      CF03(03 'SALIR')
A          R P01
A                                      ENTFLDATR
A                                      PULLDOWN(*SLTIND *RSTCSR)
A            OPCP01         2Y 0B  1  2SNGCHCFLD(*RSTCSR *SLTIND *AUTOSLT -
A                                       (*NUMCOL 1) *AUTOENT)
A                                      CHOICE(1 '>Listado Clientes')
A                                      CHOICE(2 '>WRKSPLF')
A                                      CHOICE(3 '>WRKUSRJOB')
A                                      CHOICE(4 '>WRKJOB')
A                                      CHOICE(5 '>Salir')
A          R M01
A                                      OVERLAY
A                                      MNUBAR
A                                      MNUBARDSP
A            MNB001         2Y 0B  1  2MNUBARCHC(1 P01 'Opciones Principal-
A                                      es')
```

NO COPIAR

Creación Pantalla Visualizar Mensajes (VENMSG)

```
A                                      DSPSIZ(24 80 *DS3)
A                                      INDARA
A                                      CF03(03 'Salir')
A          R WS01                      SFL
A            MENSAJE       50   B  2  6
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A          R WC01                      SFLCTL(WS01)
A                                      SFLCSRRRN(&REGREL)
A  30                                  SFLDSP
A  31                                  SFLDSPCTL
A  32                                  SFLCLR
A  30                                  SFLEND(*SCRBAR *SCRBAR)
A                                      SFLSIZ(0050)
A                                      SFLPAG(0003)
A                                      WINDOW(10 10 5 60)
A                                      WDWTITLE((*TEXT 'F3-SALIR') (*COLOR-
A                                       WHT) *BOTTOM)
A            REGREL         5S 0H
A            WDWTIT        30A  B  1 16DSPATR(HI)
A  99                                  DSPATR(UL)
A                                      DSPATR(PR)
A          R XX
A                                      KEEP
A                                      ASSUME
A                                      OVERLAY
A                                   1  3' '
```

NO COPIAR

NO COPIAR

Creación Visualizar Mensajes (VERMSJ.CBL)

```
 IDENTIFICATION DIVISION.
*-----------------------*
 PROGRAM-ID. VERMSJ.
 AUTHOR. JORGE DUARTE U.
 ENVIRONMENT DIVISION.
*--------------------*
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.      IBM-AS400.
 OBJECT-COMPUTER.      IBM-AS400.
 SPECIAL-NAMES.        LOCAL-DATA IS LOCAL-DATA-AREA .
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.


     SELECT PANTALLA   ASSIGN TO WORKSTATION-VENMSG-SI
                       ORGANIZATION IS TRANSACTION
                       ACCESS MODE  IS DYNAMIC
                       RELATIVE KEY IS NREL
                       CONTROL-AREA IS AREA-CONTROL
                       FILE STATUS  IS FS-PAN.

 DATA DIVISION.
*--------------*
 FILE SECTION.
*-------------*

 FD PANTALLA  LABEL RECORD IS OMITTED.
 01 R-PANTALLA              PIC X(1920).

 WORKING-STORAGE SECTION.
*-----------------------*
 77 I-ON                 PIC 1 VALUE B"1".
 77 I-OFF                PIC 1 VALUE B"0".
 77 I                    PIC 9(02).

 01 AREA-CONTROL.
      06 MDTO                PIC X(02).
         88 F3                VALUE "03".

 01 VARIABLES-DE-STATUS.
    05 FS-CLIENTE     PIC XX VALUE "00".
    05 FS-PAN         PIC XX VALUE "00".

 01 VARIABLES-RELATIVAS.
    05 NREL           PIC  9(03).

 01 WC01-OO
*----------
     COPY DDS-WC01-O       OF VENMSG.

 01 WS01-OO
*----------
     COPY DDS-WS01-O       OF VENMSG.

 01 WC01-IND.
*------------
    COPY DDS-WC01-INDIC    OF VENMSG.

 LINKAGE SECTION.
 01 INP-MSG.
    05 TIT-MSG           PIC X(30).
```

NO COPIAR

```
    05 ARR-MSG OCCURS 50 TIMES.
       10 MSG            PIC X(50).

 PROCEDURE DIVISION USING INP-MSG.
*---------------------------------
     OPEN I-O PANTALLA
     MOVE I-OFF     TO IN03 OF WC01-I-INDIC

     PERFORM LIMPIAR-WS01
     PERFORM CARGAR-WS01
     MOVE TIT-MSG  TO WDWTIT OF WC01-OO
     PERFORM PROCESAR-PANTALLA UNTIL
             IN03 OF WC01-I-INDIC = I-ON

     CLOSE    PANTALLA

     GOBACK.

*
 PROCESAR-PANTALLA.
*------------------
     WRITE R-PANTALLA  FROM WC01-OO FORMAT "WC01"
                       INDICATORS ARE WC01-O-INDIC END-WRITE
     READ  PANTALLA    INTO WC01-OO FORMAT "WC01"
                       INDICATORS ARE WC01-I-INDIC END-READ

     .

 LIMPIAR-WS01.
*-------------
     MOVE ZEROES          TO NREL
     MOVE I-OFF           TO IN30 OF WC01-O-INDIC
                             IN31 OF WC01-O-INDIC
     MOVE I-ON            TO IN32 OF WC01-O-INDIC
     WRITE R-PANTALLA  FROM WC01-OO FORMAT "WC01"
                       INDICATORS ARE WC01-O-INDIC END-WRITE.


 CARGAR-WS01.
*------------
     PERFORM VARYING I FROM 1 BY 1 UNTIL I > 50
        IF MSG(I) NOT = SPACES
           ADD 1   TO NREL
           MOVE MSG(I)         TO MENSAJE OF WS01-OO
           WRITE SUBFILE R-PANTALLA  FROM WS01-OO FORMAT "WS01"
                 END-WRITE
        END-IF
     END-PERFORM
     IF NREL > 0
        MOVE I-ON         TO IN30 OF WC01-O-INDIC
                             IN31 OF WC01-O-INDIC
        MOVE I-OFF        TO IN32 OF WC01-O-INDIC
     ELSE
        MOVE I-OFF        TO IN30 OF WC01-O-INDIC
        MOVE I-ON         TO IN31 OF WC01-O-INDIC
        MOVE I-OFF        TO IN32 OF WC01-O-INDIC
     END-IF
     .
```

NO COPIAR

Largo String (LARGOSTR.CBL)

```
IDENTIFICATION DIVISION.
PROGRAM-ID. LARGOSTRING.
DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.
01 VARIABLES-DE-TRABAJO.
    05 LARGO-STR PIC 9(03).

LINKAGE SECTION.
01 PAR-INPUT.
    05 STRING-CALULAR   PIC X(500).
01 PAR-OUTPUT.
    05 LK-LARGO-STR     PIC 9(03).

PROCEDURE DIVISION USING PAR-INPUT
                         PAR-OUTPUT.

    MOVE 500            TO LARGO-STR
    PERFORM UNTIL
            LARGO-STR = 0 OR
            STRING-CALULAR(LARGO-STR:1) NOT = SPACES

            SUBTRACT 1 FROM LARGO-STR

    END-PERFORM

    MOVE LARGO-STR   TO LK-LARGO-STR

    GOBACK
    .
```

NO COPIAR

Validar Rut (VALRUT.CBL)

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. VALRUT.
       DATA DIVISION.
       FILE SECTION.
       WORKING-STORAGE SECTION.
       01 VARIABLES-DE-TRABAJO.
           05 WS-RUT     PIC 9(10).
           05 WS-DV      PIC X.
           05 ENTERO     PIC 9(2).
           05 RESTO      PIC 9(2).
           05 RESTA-11   PIC 9(3).
           05 I-MULT     PIC 9.
           05 SUMA-MULT  PIC 9(5).

       LINKAGE SECTION.
       01 PARAMETRO-ENTRADA.
           05 LK-RUT    PIC 9(10).
           05 LK-DV     PIC X.
       01 PARAMETRO-SALIDA.
           05 LK-CODRET PIC 9.
      *       0 = OK
      *       1 = NO OK
       PROCEDURE DIVISION USING PARAMETRO-ENTRADA
                                PARAMETRO-SALIDA.
       MAIN-PROCEDURE.

           MOVE LK-RUT      TO WS-RUT
           MOVE 2           TO I-MULT
           MOVE 0           TO SUMA-MULT
           PERFORM UNTIL WS-RUT = 0
               DIVIDE WS-RUT BY 10 GIVING WS-RUT
                      REMAINDER RESTO
               IF I-MULT > 7
                  MOVE 2   TO I-MULT
               END-IF
               COMPUTE SUMA-MULT = SUMA-MULT + (RESTO * I-MULT)
               ADD 1        TO I-MULT
           END-PERFORM

           DIVIDE SUMA-MULT BY 11 GIVING ENTERO
                                  REMAINDER RESTO
           COMPUTE RESTA-11 = 11 - RESTO
           EVALUATE RESTA-11
             WHEN 0   MOVE "0"   TO WS-DV
             WHEN 1   MOVE "1"   TO WS-DV
             WHEN 2   MOVE "2"   TO WS-DV
             WHEN 3   MOVE "3"   TO WS-DV
             WHEN 4   MOVE "4"   TO WS-DV
             WHEN 5   MOVE "5"   TO WS-DV
             WHEN 6   MOVE "6"   TO WS-DV
             WHEN 7   MOVE "7"   TO WS-DV
             WHEN 8   MOVE "8"   TO WS-DV
             WHEN 9   MOVE "9"   TO WS-DV
             WHEN 10  MOVE "K"   TO WS-DV
             WHEN 11  MOVE "0"   TO WS-DV
           END-EVALUATE

           IF LK-DV NOT = WS-DV THEN
               MOVE 1     TO LK-CODRET
           ELSE
               MOVE 0     TO LK-CODRET
```

NO COPIAR

```
END-IF

GOBACK.
```

NO COPIAR

Validar Fecha (VALFEC.CBL)

```
******************************************************************
* AUTHOR: JORGE DUARTE
* DATE: 07-07-2025
* PURPOSE: ENSE#ANZA
* TECTONICS: COBC
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. VALFEC.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
 01 VARIABLES-DE-TRABAJO.
     05 ANO            PIC 9(04).
     05 MES            PIC 9(02).
     05 DIA            PIC 9(02).

     05 ENTERO         PIC 9(04).
     05 ENTERO-2       PIC 9(04).
     05 ENTERO-3       PIC 9(04).

     05 RESTO          PIC 9(04).
     05 RESTO-2        PIC 9(04).
     05 RESTO-3        PIC 9(04).

     05 TABLA-DIAS-MESES PIC X(24) VALUE
                       "312831303130313130313032".
     05 ARR-DIAS REDEFINES TABLA-DIAS-MESES
                OCCURS 12 TIMES.
        10 DIA-MES     PIC 9(02).

 LINKAGE SECTION.
 01 PARAMETROS-ENTRADA.
     05 FECHA-YYYYMMDD PIC X(08).
     05 FECHA-FORMATO  PIC X.
*       1 : YYYYMMDD
*       2 : DDMMYYYY

 01 PARAMETRO-SALIDA.
     05 FECHA-VALIDA   PIC X.
*       S : FECHA ES VALIDATE
*       N : FECHA ES INVALIDA

  PROCEDURE DIVISION USING PARAMETROS-ENTRADA
                           PARAMETRO-SALIDA.
     MOVE "S"            TO FECHA-VALIDA
     EVALUATE FECHA-FORMATO
      WHEN "1"
           MOVE FECHA-YYYYMMDD(1:4) TO ANO
           MOVE FECHA-YYYYMMDD(5:2) TO MES
           MOVE FECHA-YYYYMMDD(7:2) TO DIA
       WHEN "2"
           MOVE FECHA-YYYYMMDD(5:4) TO ANO
           MOVE FECHA-YYYYMMDD(3:2) TO MES
           MOVE FECHA-YYYYMMDD(1:2) TO DIA
       WHEN OTHER
           MOVE "N"         TO FECHA-VALIDA
     END-EVALUATE
     IF FECHA-VALIDA NOT = "N" THEN
         PERFORM VALIDAR-FECHA
     END-IF
```

NO COPIAR

```
     GOBACK.

 VALIDAR-FECHA.

     IF MES NOT = 2 THEN
        IF MES < 1 OR > 12 THEN
           MOVE "N"   TO FECHA-VALIDA
        ELSE
            IF ANO < 1 THEN
               MOVE "N"   TO FECHA-VALIDA
            ELSE
                IF DIA < 1 OR > DIA-MES(MES) THEN
                   MOVE "N" TO FECHA-VALIDA
                END-IF
            END-IF
        END-IF
     ELSE
        PERFORM BISIESTO

*    VERIFICAMOS SI EL MES DEL A#O ES BISIESTO
        IF (RESTO = 0 AND RESTO-2 NOT = 0) OR (RESTO-3 = 0) THEN
           MOVE 29    TO DIA-MES(MES)
        ELSE
           MOVE 28    TO DIA-MES(MES)
        END-IF
        IF DIA < 1 OR > DIA-MES(MES) THEN
           MOVE "N" TO FECHA-VALIDA
        END-IF
     END-IF
     .
 BISIESTO.
     DIVIDE ANO BY 4 GIVING ENTERO
                 REMAINDER RESTO
     DIVIDE ANO BY 100 GIVING ENTERO-2
                 REMAINDER RESTO-2
     DIVIDE ANO BY 400 GIVING ENTERO-3
                 REMAINDER RESTO-3
     .
```

NO COPIAR

Grabar Cliente (GRABARCLI.CBL)

```
 IDENTIFICATION DIVISION.
*-----------------------*
 PROGRAM-ID. GRABARCLI.
 AUTHOR. JORGE DUARTE U.
 ENVIRONMENT DIVISION.
*--------------------*
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.      IBM-AS400.
 OBJECT-COMPUTER.      IBM-AS400.
 SPECIAL-NAMES.        LOCAL-DATA IS LOCAL-DATA-AREA .
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

     SELECT CLIENTE    ASSIGN        TO DATABASE-CLIENTES
                       ORGANIZATION  IS INDEXED
                       ACCESS MODE   IS DYNAMIC
                       RECORD KEY    IS EXTERNALLY-DESCRIBED-KEY
                       FILE STATUS   IS FS-CLIENTE.


 DATA DIVISION.
*--------------*
 FILE SECTION.
*-------------*
 FD  CLIENTE   LABEL   RECORD STANDARD.
 01  R-CLIENTE.
     COPY  DDS-ALL-FORMAT  OF CLIENTES.


 WORKING-STORAGE SECTION.
*-----------------------*

 01 VARIABLES-DE-STATUS.
    05 FS-CLIENTE     PIC XX VALUE "00".

 LINKAGE SECTION.
 01 INP-REGCLI.
     COPY  DDS-ALL-FORMAT  OF CLIENTES.
 01 OUT-REGCLI.
     05 ST-REGCLI   PIC X(02).
*      '00' = OK
*    <> '00' = NO OK

 PROCEDURE DIVISION USING INP-REGCLI OUT-REGCLI.
*----------------------------------------------
     OPEN I-O CLIENTE

     MOVE CORR CLIREG OF INP-REGCLI     TO
               CLIREG OF R-CLIENTE

     WRITE R-CLIENTE END-WRITE

     MOVE FS-CLIENTE   TO ST-REGCLI

     CLOSE    CLIENTE

     GOBACK.
```

NO COPIAR

Borrar Cliente (BORRARCLI.CBL)

```
 IDENTIFICATION DIVISION.
*-----------------------*
 PROGRAM-ID. BORRARCLI.
 AUTHOR. JORGE DUARTE U.
 ENVIRONMENT DIVISION.
*--------------------*
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.      IBM-AS400.
 OBJECT-COMPUTER.      IBM-AS400.
 SPECIAL-NAMES.        LOCAL-DATA IS LOCAL-DATA-AREA .
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

     SELECT CLIENTE    ASSIGN        TO DATABASE-CLIENTES
                       ORGANIZATION  IS INDEXED
                       ACCESS MODE   IS DYNAMIC
                       RECORD KEY    IS EXTERNALLY-DESCRIBED-KEY
                       FILE STATUS   IS FS-CLIENTE.


 DATA DIVISION.
*--------------*
 FILE SECTION.
*-------------*
 FD  CLIENTE   LABEL   RECORD STANDARD.
 01  R-CLIENTE.
     COPY  DDS-ALL-FORMAT  OF CLIENTES.


 WORKING-STORAGE SECTION.
*-----------------------*

 01 VARIABLES-DE-STATUS.
    05 FS-CLIENTE     PIC XX VALUE "00".

 LINKAGE SECTION.
 01 INP-REGCLI.
     05 LK-RUT      PIC 9(10).
 01 OUT-REGCLI.
     05 ST-REGCLI   PIC X(02).
*      '00' = OK
*    <> '00' = NO OK

 PROCEDURE DIVISION USING INP-REGCLI OUT-REGCLI.
*----------------------------------------------
     OPEN I-O CLIENTE

     MOVE LK-RUT     TO RUT OF R-CLIENTE
     READ CLIENTE END-READ
     IF FS-CLIENTE = "00"
        DELETE CLIENTE END-DELETE
     END-IF
     MOVE FS-CLIENTE   TO ST-REGCLI

     CLOSE    CLIENTE

     GOBACK.
```

NO COPIAR

NO COPIAR

Mantenedor de Cliente (MANCLICBL.CBL)

```
 IDENTIFICATION DIVISION.
*-----------------------*
 PROGRAM-ID. MANCLICBL.
 AUTHOR. JORGE DUARTE U.
 ENVIRONMENT DIVISION.
*--------------------*
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.      IBM-AS400.
 OBJECT-COMPUTER.      IBM-AS400.
 SPECIAL-NAMES.        LOCAL-DATA IS LOCAL-DATA-AREA .
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

     SELECT PANTALLA   ASSIGN TO WORKSTATION-MANT01-SI
                       ORGANIZATION IS TRANSACTION
                       ACCESS MODE  IS DYNAMIC
                       RELATIVE KEY IS NREL
                       FILE STATUS  IS FS-PAN.

 DATA DIVISION.
*--------------*
 FILE SECTION.
*-------------*
 FD PANTALLA  LABEL RECORD IS OMITTED.
 01 R-PANTALLA             PIC X(1920).

 WORKING-STORAGE SECTION.
*-----------------------*
 77 I-ON                  PIC 1 VALUE B"1".
 77 I-OFF                 PIC 1 VALUE B"0".
 77 I-MSG                 PIC 9(2) VALUE ZEROES.

 01 AREA-CONTROL.
       06 MDTO                PIC X(02).
          88 F3               VALUE "03".
          88 F7               VALUE "07".

 01 VARIABLES-DE-TRABAJO.
     05 WS-EMAIL       PIC X(100).
     05 CUENTA-1          PIC 9(03).
     05 CUENTA-2          PIC 9(03).
     05 CUENTA-3          PIC 9(03).
     05 WS-ANTES-DE-ARROBA   PIC X(100).
     05 WS-DESPUES-DE-ARROBA PIC X(100).

 01 VARIABLES-DE-STATUS.
    05 FS-PAN          PIC XX VALUE "00".

 01 VARIABLES-RELATIVAS.
    05 NREL            PIC 9(03).

 01 R01-II.
*----------
     COPY DDS-R01-I        OF MANT01.

 01 R01-INDIC.
*--------------
     COPY DDS-R01-INDIC    OF MANT01.

 01 INP-MSG.
    05 TIT-MSG           PIC X(30).
```

NO COPIAR

```
    05 ARR-MSG OCCURS 50 TIMES.
       10 MSG            PIC X(50).

 01 VALRUT-ENTRADA.
     05 VAL-RUT   PIC 9(10).
     05 VAL-DV    PIC X.
 01 VALRUT-SALIDA.
     05 VAL-CODRET PIC 9.
*       0 = OK
*       1 = NO OK

 01 VALFEC-ENTRADA.
     05 FECHA-YYYYMMDD PIC X(08).
     05 FECHA-FORMATO  PIC X.
*       1 : YYYYMMDD
*       2 : DDMMYYYY

 01 VALFEC-SALIDA.
     05 FECHA-VALIDA   PIC X.
*       S : FECHA ES VALIDATE
*       N : FECHA ES INVALIDA

 01 PARAMETROS-LARGO-STRING.
     05 STRING-CALCULAR   PIC X(500).
     05 STRING-LARGO      PIC 9(03).

 01 INP-REGCLI.
     COPY  DDS-ALL-FORMAT  OF CLIENTES.
 01 OUT-REGCLI.
     05 ST-REGCLI   PIC X(02).
*       '00' = OK
*    <> '00' = NO OK

 PROCEDURE DIVISION.
*--------------------
     OPEN I-O  PANTALLA
     MOVE I-OFF       TO IN03 OF R01-I-INDIC
     INITIALIZE R01-I REPLACING ALPHANUMERIC DATA BY SPACES
                                NUMERIC DATA BY ZEROES
     PERFORM PROCESAR-PANTALLA UNTIL
             IN03 OF R01-I-INDIC = I-ON

     CLOSE     PANTALLA

     GOBACK.

*
 PROCESAR-PANTALLA.
*------------------
     WRITE R-PANTALLA  FROM R01-II  FORMAT "R01"
                       INDICATORS ARE R01-O-INDIC  END-WRITE
     READ  PANTALLA    INTO R01-II  FORMAT "R01"
                       INDICATORS ARE R01-I-INDIC  END-READ

     MOVE ZEROES         TO I-MSG
     IF IN03 OF R01-I-INDIC = I-OFF AND
        IN07 OF R01-I-INDIC = I-OFF
        PERFORM VALIDA-CAMPOS
        MOVE I-OFF     TO IN03 OF R01-I-INDIC
     ELSE
        IF IN07 OF R01-I-INDIC = I-ON
           PERFORM GRABA-REGISTRO
           MOVE I-OFF    TO IN50 OF R01-O-INDIC
```

NO COPIAR

```
        END-IF
     END-IF
     IF I-MSG > 0 THEN
        CALL "VERMSJ" USING INP-MSG
     END-IF
     .

 VALIDA-CAMPOS.
*--------------
     PERFORM LIMPIA-ARR-MSG
     MOVE ZEROES         TO I-MSG

     PERFORM VALIDA-RUT
     PERFORM VALIDA-NOMBRE-APELLIDOS
     PERFORM VALIDA-FECHA-NACIMIENTO
     PERFORM VALIDA-GENERO
     PERFORM VALIDA-DIRECCION
     PERFORM VALIDA-CIUDAD
     PERFORM VALIDA-TELEFONOS
     PERFORM VALIDA-EMAIL
     IF I-MSG = 0 THEN
        MOVE I-ON   TO IN50 OF R01-O-INDIC
     END-IF

     .

 LIMPIA-ARR-MSG.

     PERFORM VARYING I-MSG FROM 1 BY 1 UNTIL I-MSG > 50
        MOVE SPACES TO MSG(I-MSG)
     END-PERFORM
     .

 VALIDA-RUT.
*-----------
     IF RUT OF R01-II = ZEROES       OR
        DV  OF R01-II = SPACES      THEN
         ADD  1               TO I-MSG
         MOVE "RUT INCORRECTO" TO MSG(I-MSG)
     ELSE
         MOVE RUT OF R01-II TO VAL-RUT
         MOVE DV  OF R01-II TO VAL-DV
         MOVE 0          TO VAL-CODRET
         CALL "VALRUT"    USING VALRUT-ENTRADA
                                VALRUT-SALIDA
         IF VAL-CODRET = 1 THEN
            ADD   1               TO I-MSG
            MOVE "RUT INCORRECTO" TO MSG(I-MSG)
         END-IF
     END-IF
     .


 VALIDA-NOMBRE-APELLIDOS.
*------------------------
     IF NOMBRE OF R01-II = SPACES  OR
        APEPAT OF R01-II = SPACES  OR
        APEMAT OF R01-II = SPACES
        ADD  1                     TO I-MSG
        MOVE "NOMBRE O APELLIDOS INCORRECTOS"
                                   TO MSG(I-MSG)
     END-IF
     .
```

NO COPIAR

```
 VALIDA-FECHA-NACIMIENTO.
*-----------------------
     MOVE FECNAC OF R01-II TO FECHA-YYYYMMDD
     MOVE "2"              TO FECHA-FORMATO

     MOVE " "           TO FECHA-VALIDA
     CALL "VALFEC"       USING VALFEC-ENTRADA
                               VALFEC-SALIDA

     IF FECHA-VALIDA = "N" THEN
        ADD  1                    TO I-MSG
        MOVE "FECHA NACIMIENTO INCORRECTA"
                                  TO MSG(I-MSG)
     END-IF
     .

 VALIDA-GENERO.
*--------------
     IF GENERO OF R01-II = SPACES  OR
        (GENERO OF R01-II NOT = "H" AND "M" )         THEN

        ADD  1                    TO I-MSG
        MOVE "GÉNERO DEBE SER H Ó M"
                                  TO MSG(I-MSG)
     END-IF
     .

 VALIDA-DIRECCION.
*-----------------
     MOVE DIRECC OF R01-II TO STRING-CALCULAR
     PERFORM CALCULAR-LARGO-STRING
     IF STRING-LARGO < 5 THEN
        ADD  1                TO I-MSG
        MOVE "DEBE INGRESAR UNA DIRECCION MINIMO 5 CARACTERES"
                              TO MSG(I-MSG)
     END-IF
     .

 VALIDA-CIUDAD.
*--------------
     MOVE CIUDAD OF R01-II TO STRING-CALCULAR
     PERFORM CALCULAR-LARGO-STRING
     IF STRING-LARGO < 5 THEN
        ADD  1                TO I-MSG
        MOVE "DEBE INGRESAR UNA CIUDAD    MINIMO 5 CARACTERES"
                              TO MSG(I-MSG)
     END-IF
     .
 VALIDA-TELEFONOS.
*-----------------
     IF TELEFONO1 OF R01-II = SPACES AND
        TELEFONO2 OF R01-II = SPACES
        ADD  1                TO I-MSG
        MOVE "DEBE INGRESAR AL MENOS 1 TELEFONO"
                              TO MSG(I-MSG)
     END-IF
     .

 VALIDA-EMAIL.
*-------------
     IF EMAIL OF R01-II = SPACES THEN
        ADD  1                TO I-MSG
```

NO COPIAR

```
        MOVE "DEBE INGRESAR EMAIL              "
                                TO MSG(I-MSG)
      ELSE
       MOVE EMAIL OF R01-II     TO WS-EMAIL
       MOVE ZEROES              TO CUENTA-1
                                   CUENTA-2
                                   CUENTA-3
       INSPECT WS-EMAIL TALLYING CUENTA-1 FOR ALL "@"
       IF CUENTA-1 = 1 THEN
            UNSTRING WS-EMAIL DELIMITED BY "@"
            INTO     WS-ANTES-DE-ARROBA,
                     WS-DESPUES-DE-ARROBA
            IF WS-ANTES-DE-ARROBA(1:3) NOT = SPACES THEN
               INSPECT WS-DESPUES-DE-ARROBA TALLYING
               CUENTA-3 FOR ALL "."
               IF CUENTA-3 = 0 THEN
                  ADD 1         TO I-MSG
                  MOVE
                  "DEBE INGRESAR EMAIL CON FORMATO CORRECTO"
                                 TO MSG(I-MSG)
               ELSE
                   IF WS-DESPUES-DE-ARROBA(1:3) = SPACES THEN
                      ADD 1     TO I-MSG
                      MOVE
                     "DEBE INGRESAR EMAIL CON FORMATO CORRECTO"
                      TO MSG(I-MSG)
                   END-IF
               END-IF
            ELSE
              ADD 1     TO I-MSG
              MOVE "DEBE INGRESAR EMAIL CON FORMATO CORRECTO    "
                                    TO MSG(I-MSG)
            END-IF

        ELSE
           ADD 1     TO I-MSG
           MOVE "DEBE INGRESAR EMAIL CON FORMATO CORRECTO (5)"
                                    TO MSG(I-MSG)
         END-IF
      END-IF
      .
*
 CALCULAR-LARGO-STRING.
*----------------------
      MOVE 0              TO STRING-LARGO
      CALL "LARGOSTR"     USING STRING-CALCULAR STRING-LARGO
      .

*
 GRABA-REGISTRO.
*---------------
       INITIALIZE CLIREG OF INP-REGCLI
                  REPLACING ALPHANUMERIC DATA BY SPACES
                                NUMERIC DATA BY ZEROES
       MOVE CORR R01-I TO CLIREG OF INP-REGCLI
       CALL "GRABARCLI"  USING INP-REGCLI
                               OUT-REGCLI
       IF ST-REGCLI = "00"
          ADD 1           TO I-MSG
          MOVE "REGISTRO AGREGADO" TO MSG(I-MSG)
       ELSE
          ADD 1           TO I-MSG
          STRING "REGISTRO NO INSERTADO " DELIMITED BY SIZE
```

NO COPIAR

```
           " STATUS ("              DELIMITED BY SIZE
           ST-REGCLI                DELIMITED BY SIZE
           ") "                     DELIMITED BY SIZE
                                INTO MSG(I-MSG)
END-IF
.
```

NO COPIAR

Listado de Cliente (LISCLI.CBL)

```
 IDENTIFICATION DIVISION.
*-----------------------*
 PROGRAM-ID. LISCLI.
 AUTHOR. JORGE DUARTE U.
 ENVIRONMENT DIVISION.
*--------------------*
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.      IBM-AS400.
 OBJECT-COMPUTER.      IBM-AS400.
 SPECIAL-NAMES.        LOCAL-DATA IS LOCAL-DATA-AREA .
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

     SELECT CLIENTE    ASSIGN        TO DATABASE-CLIENTES
                       ORGANIZATION  IS INDEXED
                       ACCESS MODE   IS DYNAMIC
                       RECORD KEY    IS EXTERNALLY-DESCRIBED-KEY
                       FILE STATUS   IS FS-CLIENTE.


     SELECT PANTALLA   ASSIGN TO WORKSTATION-LIST01-SI
                       ORGANIZATION IS TRANSACTION
                       ACCESS MODE  IS DYNAMIC
                       RELATIVE KEY IS NREL
                       CONTROL-AREA IS AREA-CONTROL
                       FILE STATUS  IS FS-PAN.

 DATA DIVISION.
*--------------*
 FILE SECTION.
*-------------*

 FD  CLIENTE   LABEL   RECORD STANDARD.
 01  R-CLIENTE.
     COPY  DDS-ALL-FORMAT  OF CLIENTES.

 FD PANTALLA  LABEL RECORD IS OMITTED.
 01 R-PANTALLA            PIC X(1920).

 WORKING-STORAGE SECTION.
*-----------------------*
 77 I-ON                  PIC 1 VALUE B"1".
 77 I-OFF                 PIC 1 VALUE B"0".
 77 I                     PIC 9(02).

 01 AREA-CONTROL.
       06 MDTO               PIC X(02).
          88 F3              VALUE "03".

 01 VARIABLES-DE-STATUS.
    05 FS-CLIENTE     PIC XX VALUE "00".
    05 FS-PAN         PIC XX VALUE "00".

 01 VARIABLES-RELATIVAS.
    05 NREL           PIC  9(03).

 01 C01-OO.
*----------
     COPY DDS-C01-O        OF LIST01.

 01 S01-OO.
```

NO COPIAR

```
*----------
     COPY DDS-S01-O        OF LIST01.

 01 C01-IND.
*------------
    COPY DDS-C01-INDIC     OF LIST01.

 01 DEL-INPUT.
     05 DEL-RUT     PIC 9(10).
 01 DEL-OUTPUT.
     05 DEL-STATUS  PIC X(02).
*       '00' = OK
*    <> '00' = NO OK

 PROCEDURE DIVISION.
*-------------------
     OPEN I-O PANTALLA
     OPEN INPUT CLIENTE

     MOVE I-OFF     TO IN03 OF C01-I-INDIC

     MOVE ZEROES    TO RUTPOS OF C01-OO

     PERFORM CARGAR-S01
     PERFORM PROCESAR-PANTALLA UNTIL
             IN03 OF C01-I-INDIC = I-ON

     CLOSE    PANTALLA CLIENTE

     GOBACK.

*
 PROCESAR-PANTALLA.
*------------------
     WRITE R-PANTALLA              FORMAT "R01"
     WRITE R-PANTALLA  FROM C01-OO FORMAT "C01"
                       INDICATORS ARE C01-O-INDIC END-WRITE
     READ  PANTALLA    INTO C01-OO FORMAT "C01"
                       INDICATORS ARE C01-I-INDIC END-READ

     IF IN06 OF C01-I-INDIC = I-ON
        CALL "MANCLICBL"
        CANCEL "MANCLICBL"
     ELSE
        PERFORM RECORRER-S01 UNTIL FS-PAN NOT = "00"
     END-IF
     PERFORM CARGAR-S01

     .

 LIMPIAR-S01.
*-------------
     MOVE ZEROES          TO NREL
     MOVE I-OFF           TO IN30 OF C01-O-INDIC
                             IN31 OF C01-O-INDIC
     MOVE I-ON            TO IN32 OF C01-O-INDIC
     WRITE R-PANTALLA  FROM C01-OO FORMAT "C01"
                       INDICATORS ARE C01-O-INDIC END-WRITE.


 CARGAR-S01.
*-------------
     PERFORM LIMPIAR-S01
```

NO COPIAR

```
     MOVE RUTPOS OF C01-OO   TO RUT OF CLIENTE
     START  CLIENTE KEY IS NOT < EXTERNALLY-DESCRIBED-KEY
            END-START
     IF FS-CLIENTE = "00"
        PERFORM LEER-NEXT-CLIENTE
        PERFORM RECORRE-CLIENTE
                UNTIL FS-CLIENTE NOT = "00"
                OR NREL > 100
     END-IF
     IF NREL > 0
        MOVE I-ON         TO IN30 OF C01-O-INDIC
                             IN31 OF C01-O-INDIC
        MOVE I-OFF        TO IN32 OF C01-O-INDIC
     ELSE
        MOVE I-OFF        TO IN30 OF C01-O-INDIC
        MOVE I-ON         TO IN31 OF C01-O-INDIC
        MOVE I-OFF        TO IN32 OF C01-O-INDIC
     END-IF
     .
*
 RECORRE-CLIENTE.
*----------------
     IF FS-CLIENTE = "00"
           ADD 1   TO NREL
           MOVE SPACES          TO  OPCS01 OF S01-OO
           MOVE RUT OF CLIENTE  TO  RUTS01 OF S01-OO
           MOVE DV  OF CLIENTE  TO  DVS01  OF S01-OO
           MOVE SPACES          TO  NOMS01 OF S01-OO
           STRING APEPAT OF CLIENTE   DELIMITED BY " "
                  " "                 DELIMITED SIZE
                  APEMAT OF CLIENTE   DELIMITED BY " "
                  " "                 DELIMITED SIZE
                  NOMBRE OF CLIENTE   DELIMITED BY " "
                                 INTO NOMS01 OF S01-OO
           WRITE SUBFILE R-PANTALLA  FROM S01-OO FORMAT "S01"
                 END-WRITE
     END-IF
     PERFORM LEER-NEXT-CLIENTE
     .

 LEER-NEXT-CLIENTE.
*------------------
      READ CLIENTE NEXT RECORD END-READ
     .

 RECORRER-S01.
*-------------
     READ SUBFILE PANTALLA  NEXT MODIFIED INTO S01-OO
                  FORMAT "S01"  END-READ
     IF FS-PAN  = "00"
        IF OPCS01 OF S01-OO = "B" OR "b"
           MOVE  RUTS01 OF S01-OO  TO DEL-RUT
           MOVE  SPACES            TO DEL-STATUS
           CALL "BORRARCLI" USING DEL-INPUT
                                  DEL-OUTPUT
        END-IF
        IF OPCS01 OF S01-OO = "C" OR "c"
           CALL "CONCLI"    USING RUTS01 OF S01-OO
           CANCEL "CONCLI"
        END-IF
     END-IF
     .
```

NO COPIAR

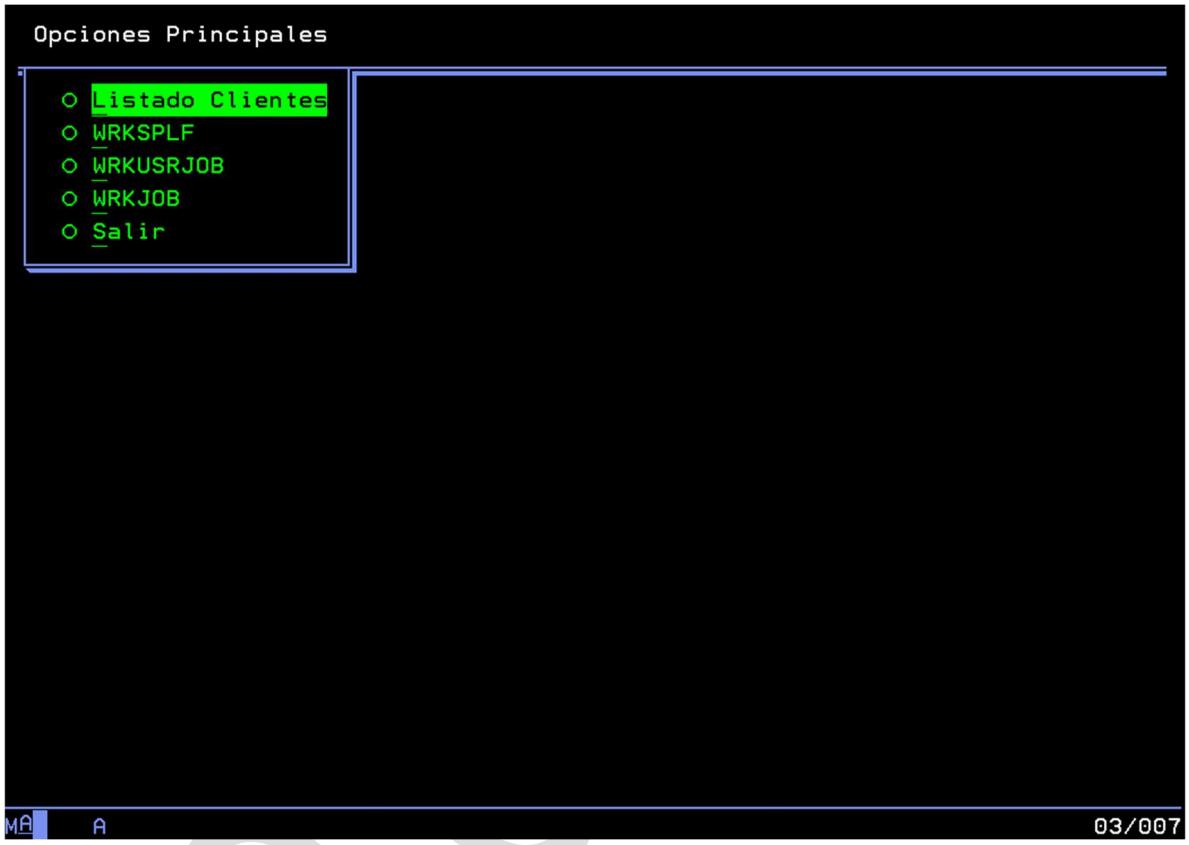
Programa Menu (MENUCLP)

```
 PGM
        DCLF       FILE(LIBJDU/MENU01) RCDFMT(M01 P01)


 INICIO:
         SNDRCVF RCDFMT(M01) WAIT(*YES)
         IF COND(&IN03 *EQ '1') THEN(DO)
            GOTO CMDLBL(FIN)
         ENDDO
         IF COND(&MNB001 *EQ 1) THEN(DO)
            GOTO CMDLBL(PP01)
         ENDDO
         GOTO CMDLBL(INICIO)
PP01:
     RCVF RCDFMT(P01) WAIT(*YES)
     IF COND(&OPCP01 *EQ 1) THEN(DO)
       CALL PGM(LISCLI)

     ENDDO
     IF COND(&OPCP01 *EQ 2) THEN(DO)
        ?WRKSPLF
     ENDDO
     IF COND(&OPCP01 *EQ 3) THEN(DO)
        ?WRKUSRJOB
     ENDDO
     IF COND(&OPCP01 *EQ 4) THEN(DO)
        ?WRKJOB
     ENDDO
     IF COND(&OPCP01 *EQ 5) THEN(DO)
        GOTO CMDLBL(FIN)
     ENDDO
     GOTO CMDLBL(INICIO)

FIN: ENDPGM
```

NO COPIAR

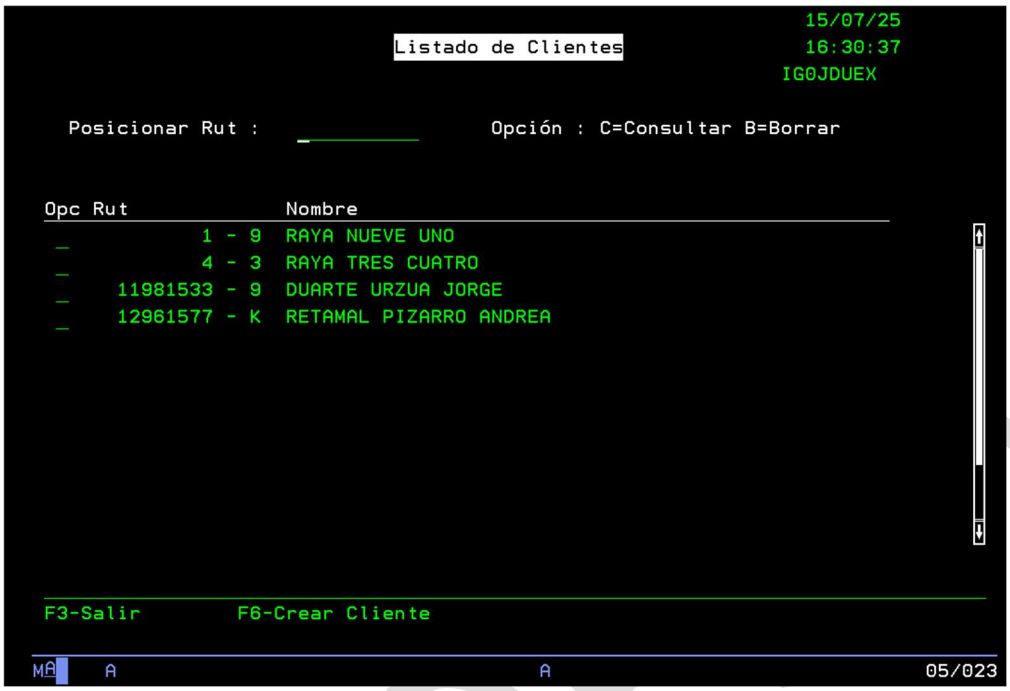
Ejecución Programas AS400

Una vez compilados todos los fuentes en la Biblioteca que se haya creado para tal efecto, procedemos a ejecutar el CL **CALL MENUCLP**

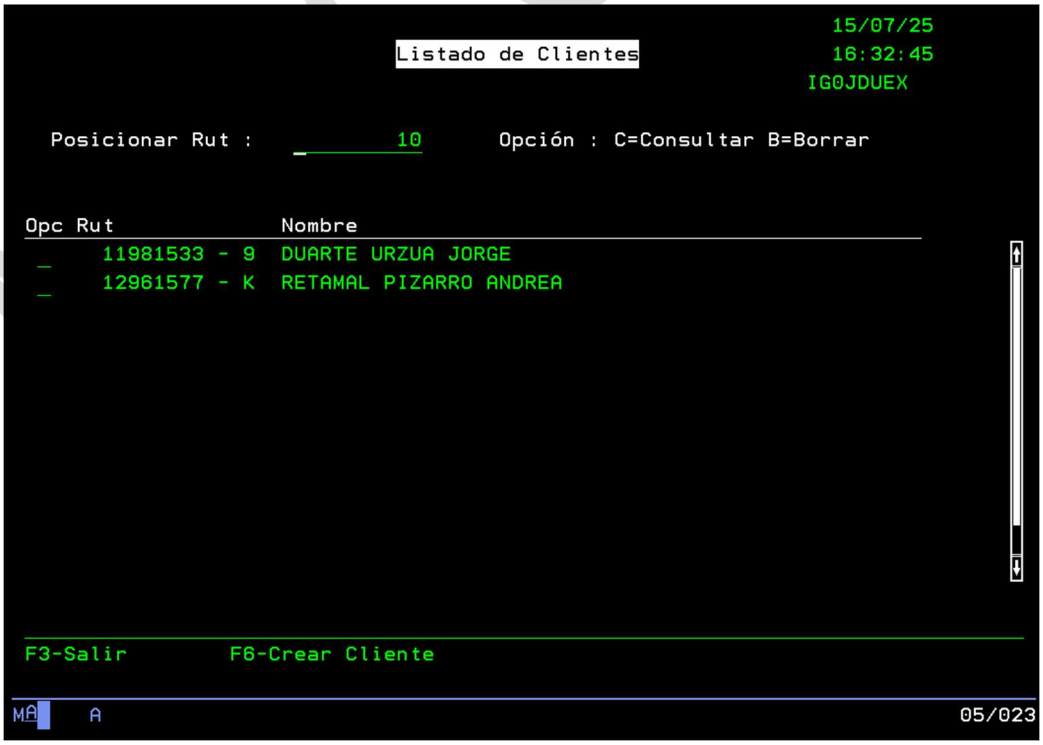


NO

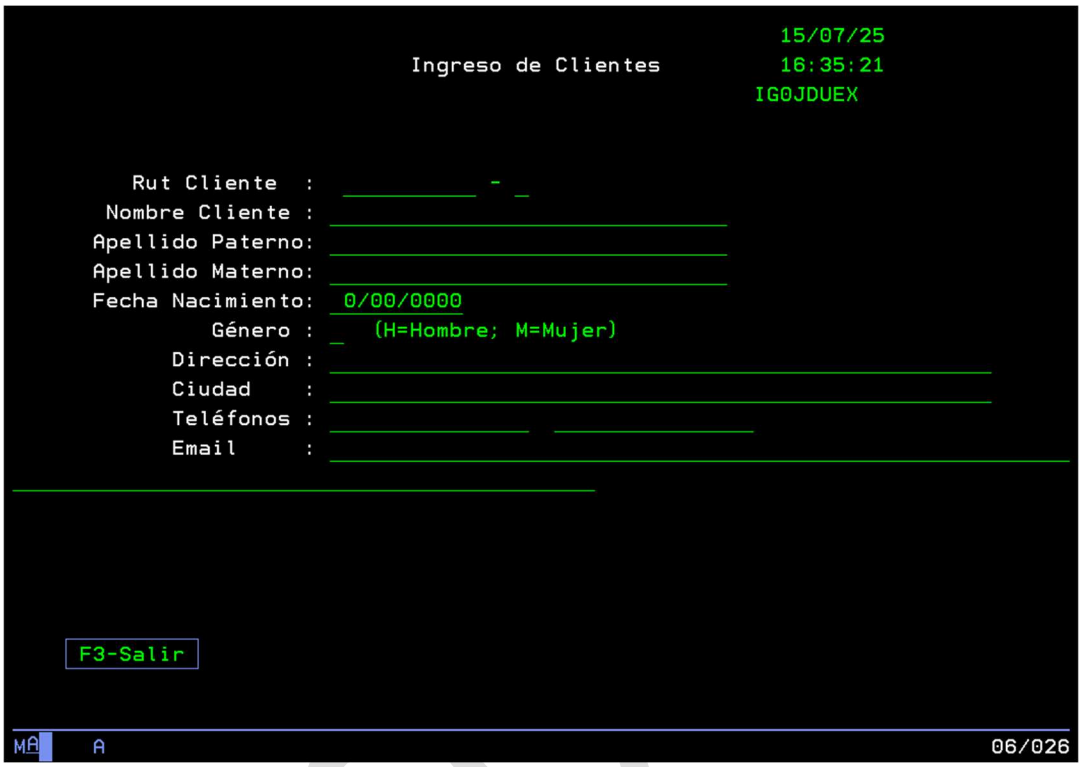
Seleccionamos la Opción Listado de Clientes, esto llamará al programa LISCLI.



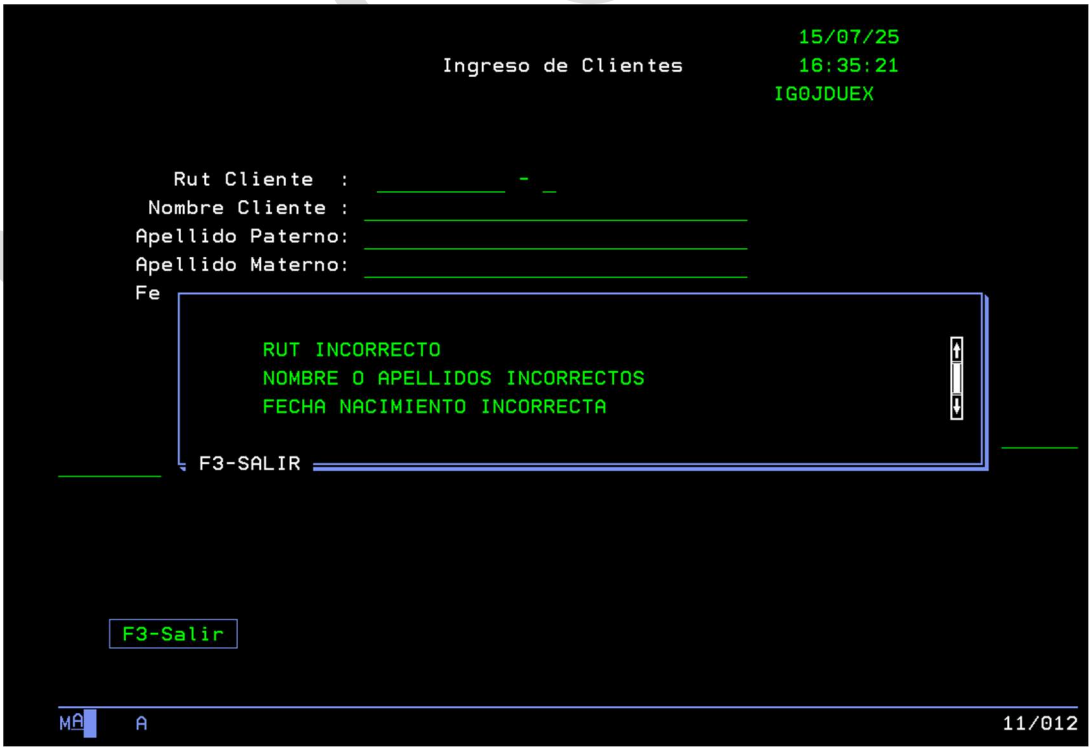
Acá al igual que en los ejemplos de la sección inicial podemos posicionar el rut, para nuestro ejemplo seleccionaremos los mayores a 10.

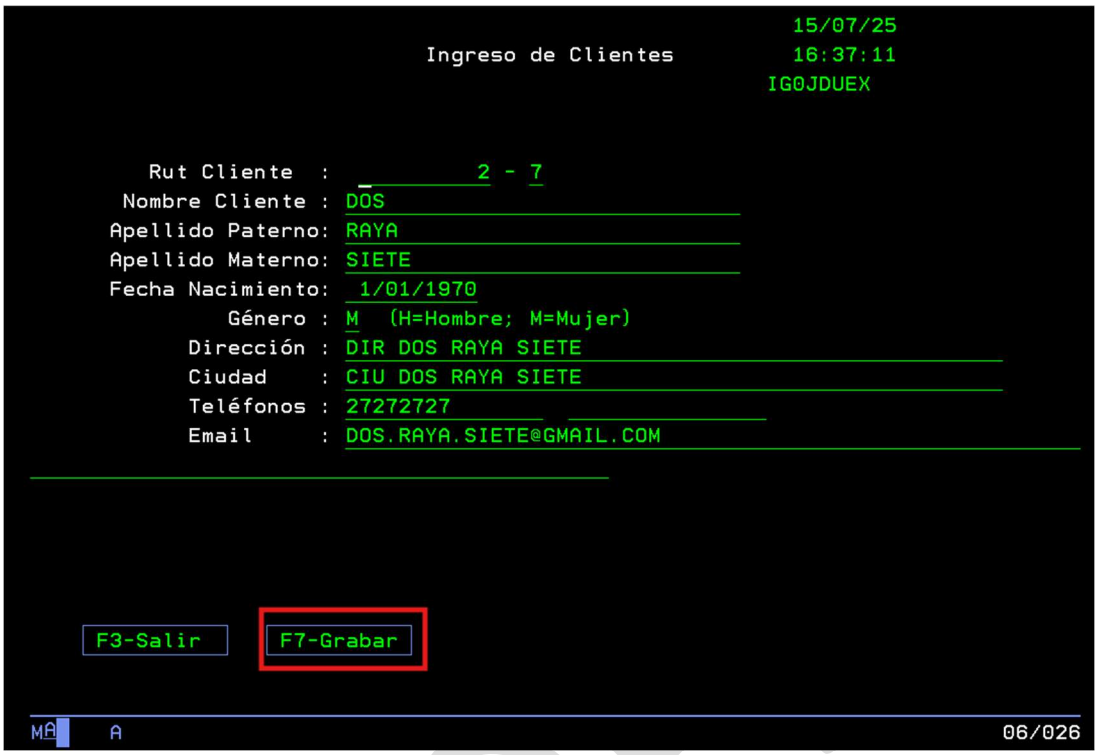


Si presionamos **F6**, ejecutaremos el Programa MANTCLI, que permite crear nuevos Clientes, además este programa usa rutinas de **VALFEC**, **VALRUT** y **VERMSJ**.

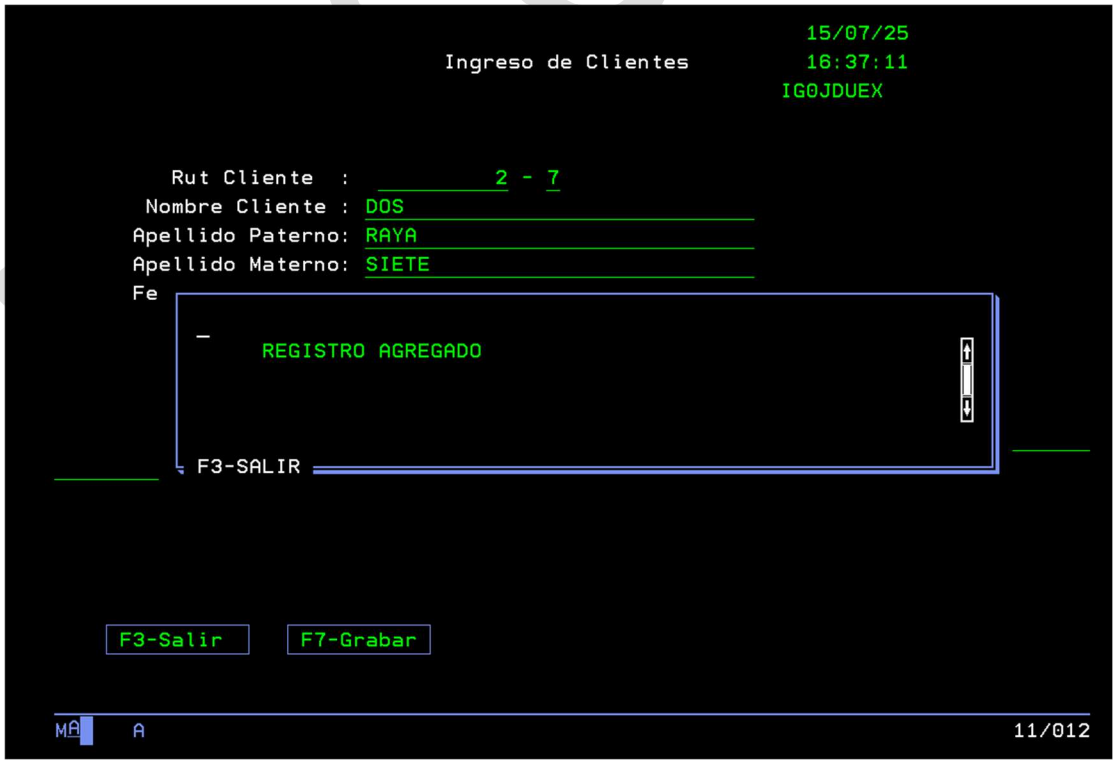


Si presionamos ENTER, se validarán los campos.

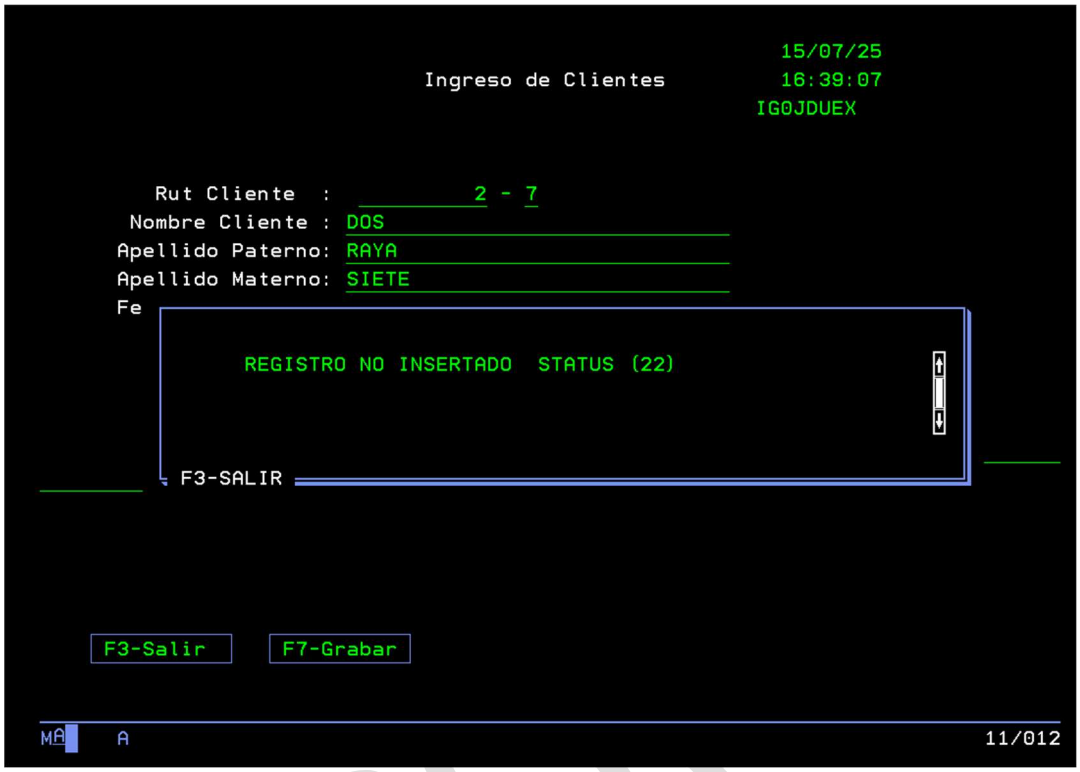




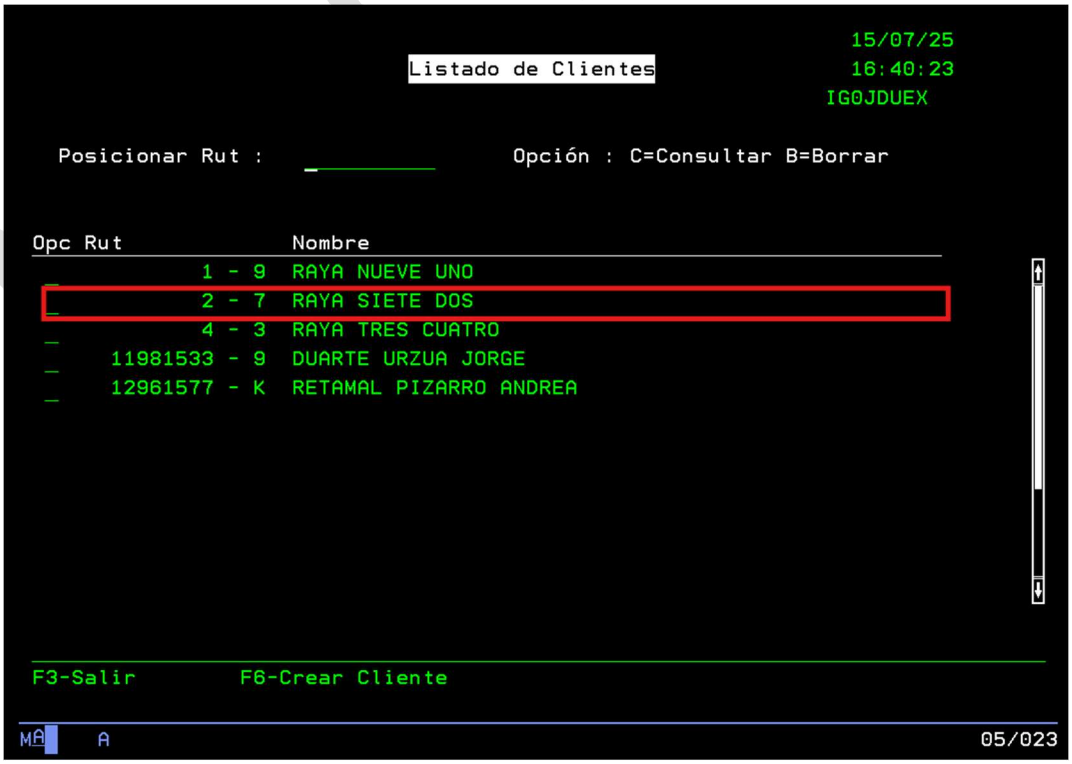
Si agregamos todos los datos y no hay errores, aparecerá la Función F7-Grabar, esto permite grabar en archivo CLIENTE la información que se ingreso, en este caso de usa la rutina GRABARCLI.



En caso que volvamos a Ingresar el mismo registro, retornará el Error.



Una vez que se vuelve al Listado de Clientes se actualiza la lista, tal como se muestra a continuación.



Opcion Borrar

Está opción utiliza la Rutina BORRARCLI, esto elimina el registro del Archivo CLIENTE.

Al volver se vuelve actualizar la lista de Clientes.

Opción Consultar, está opción permite Consultar los Datos del Cliente, para ello utiliza la rutina **CONCLI**.

Referencias

- https://blogthinkbig.com/cobol-sigue-vivo
- https://launchpad.net/cobcide/+download
- https://docs.bmc.com/xwiki/bin/view/Mainframe/Storage/BMC-Compuware-IAM/bciam10/Messages-and-Codes/COBOL-File-Status-Codes-with-IAM/
- https://www.ibm.com/docs/es/cobol-linux-x86/1.2?topic=pdf-version-documentation
- https://soloconlinux.org.es/manuales-cobol/

NO COPIAR